1. Time Validation

QUESTION:

Time Validation

Write code to validate time using the following rules:

Business rules:

- It should be a valid time in 12 hrs format
- It should have case insensitive AM or PM
-The time as input in the following format 'hh:mm am' or 'hh:mm pm'

Example:

input = 09:59 pm

output = Valid time format

Include a class UserProgramCode with static method validateTime which accepts the String.The return type should be interger.

Create a class Program which would get the input and call the static method validateTime present in the UserProgramCode.

If the given time is as per the given business rules return 1 else return -1.If the method returns 1 then print "Valid time format" else print "Invalid time format" in Program.

Input and Output Format:

The input time will be a string

Output will be a string.("Valid time format" or "Invalid time format").

Sample Input 1:

09:59 pm

Sample Output 1:

Valid time format

ANSWER:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace levelI_02
{
   class Program
   {
      static void Main(string[] args)
      {
         string str = Console.ReadLine();

         int ans = UserProgramCode.validateTime(str);

         if (ans == 1)
            Console.WriteLine("Valid time format");
         else if (ans == -1)
            Console.WriteLine("Invalid time format");
      }
   }
   class UserProgramCode
   {
      public static int validateTime(string str)
      {


         int hr, min;

         hr = int.Parse(str.Substring(0, 2));
         min = int.Parse(str.Substring(3, 2));
         string suf = str.Substring(5, 3);
```

```
        if (hr > 12 || min > 60 || suf != " am" && suf != " pm")
            return -1;
        else
            return 1;


    }
  }
}
```

2. Sum Largest Numbers In Range

QUESTION:

Sum Largest Numbers In Range

Given an array of integer as input1 which falls under the range 1-100, write a program to find the largest numbers from input1 which would fall in the given range 1-10 , 11-20, 21-30, 31-40, …… till 91-100. Now find their sum and print the sum. Business Rules: 1. If the given input array contains any negative number then print -1. 2. If any element is equal to zero or greater than 100 then print -2. 3. In case the array of integer satisfies both business rule 1 as well as 2 then print -3. 4. In case of duplicate numbers eliminate the duplicate number and follow all other steps for calculation of the largest number. Create a class named UserProgramCode that has the following static method

public static int largestNumber(int[] input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:

The first line of the input consists of an integer n, that corresponds to the size of the array.

The next 'n' lines of input consist of integers that correspond to the elements in the array.

Output is an integer.

Refer business rules and sample output for output format.

Sample Input 1: 7 13 18 26 34 58 65 54

Sample Output 1 :

201 Sample Input 2 :

5 -1 19 15 18 101

Sample Output 2 :

-3

ANSWER:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace SumLargestInRange
{
   class Program
   {
      static void Main(string[] args)
      {
         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 op=UserProgramCode.sumrange(arr);
         Console.WriteLine(op);
         Console.ReadLine();
```

```
    }
  }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace sum_Largest_in_range
{
  class UserProgramCode
  {
    public static int sumrange(int[] arr)
    {
      int li = 0, c = 0, c1 = 0, c2 = 0, c3 = 0, c4 = 0, c5 = 0, c6 = 0, c7 = 0, c8 = 0, c9 = 0;
      int s = 0;
      int count = 0;
      for (int i = 0; i < arr.Length; i++)
      {
        if ((arr[i] < 0))
        {
          count++;
        }
        if ((arr[i] == 0) || (arr[i] > 100))
        {
          count++;
        }
        if (count == 2)
          return -3;
```

```
}
for (int i = 0; i < arr.Length; i++)
{
   if ((arr[i] == 0) || (arr[i] > 100))
   {
      return -2;
   }

   else if (arr[i] < 0)
   {
      return -1;
   }



}
for (int i = 0; i < arr.Length; i++)
{
   if (arr[i] > 0 && arr[i] <= 10)
   {
      if (c == 0)
      {
         s = s + arr[i];
      }
      else
      {
         if (arr[li] < arr[i])
         {
            s = s - arr[li];
            s = s + arr[i];
         }
      }
      c++;
```

```
    li = i;
}

else if (arr[i] > 10 && arr[i] <= 20)
{
    if (c1 == 0)
    {
        li = 0;
        s = s + arr[i];
    }
    else
    {
        if (arr[li] < arr[i])
        {
            s = s - arr[li];
            s = s + arr[i];
        }
    }
    c1++;
    li = i;
}
else if (arr[i] > 20 && arr[i] <= 30)
{
    if (c2 == 0)
    {
        li = 0;
        s = s + arr[i];
    }
    else
    {
        if (arr[li] < arr[i])
        {
```

```
            s = s - arr[li];
            s = s + arr[i];
        }
    }
    c2++;
    li = i;
}
else if (arr[i] > 30 && arr[i] <= 40)
{
    if (c3 == 0)
    {
        li = 0;
        s += arr[i];
    }
    else
    {
        if (arr[li] < arr[i])
        {
            s = s - arr[li];
            s = s + arr[i];
        }
    }
    c3++;
    li = i;
}
else if (arr[i] > 40 && arr[i] <= 50)
{
    if (c4 == 0)
    {
        li = 0;
        s = s + arr[i];
    }
```

```
    else
    {
        if (arr[li] < arr[i])
        {
            s = s - arr[li];
            s = s + arr[i];
        }
    }
    c4++;
    li = i;
}
else if (arr[i] > 50 && arr[i] <= 60)
{
    if (c5 == 0)
    {
        li = 0;
        s = s + arr[i];
    }
    else
    {
        if (arr[li] < arr[i])
        {
            s = s - arr[li];
            s = s + arr[i];
        }
    }
    c5++;
    li = i;
}
else if (arr[i] > 60 && arr[i] <= 70)
{
    if (c6 == 0)
```

```
    {
        li = 0;
        s = s + arr[i];
    }
    else
    {
        if (arr[li] < arr[i])
        {
            s = s - arr[li];
            s = s + arr[i];
        }
    }
    c6++;
    li = i;
}
else if (arr[i] > 70 && arr[i] <= 80)
{
    if (c7 == 0)
    {
        li = 0;
        s = s + arr[i];
    }
    else
    {
        if (arr[li] < arr[i])
        {
            s = s - arr[li];
            s = s + arr[i];
        }
    }
    c7++;
    li = i;
```

```
}
else if (arr[i] > 80 && arr[i] <= 90)
{
   if (c8 == 0)
   {
      li = 0;
      s = s + arr[i];
   }
   else
   {
      if (arr[li] < arr[i])
      {
         s = s - arr[li];
         s = s + arr[i];
      }
   }
   c8++;
   li = i;
}
else if (arr[i] > 90 && arr[i] <= 100)
{
   if (c9 == 0)
   {
      li = 0;
      s = s + arr[i];
   }
   else
   {
      if (arr[li] < arr[i])
      {
         s = s - arr[li];
         s = s + arr[i];
```

```
                }
            }
            c9++;
            li = i;
        }



    }



    return s;



    }
  }
}
```

3. Next Consonant or Vowel

QUESTION:

Next Consonant or Vowel

Given an input String, write a program to replace all the vowels of the given string with the next consonant and replace all consonants with the next available vowel. Business Rule: 1. If the input string contains any number or any special characters, print 'Invalid input'. 2. The input is case sensitive. Please ensure that each character in the output has exactly the same case as the input string. Create a class named UserProgramCode that has the following static method

public static string nextString(String input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:

Input consists of a string.

Output consists of a string. Refer business rules and sample output for the format.

Sample Input 1 : zebRa

Sample Output 1 : afeUb Sample Input 2 : cat@rat/123

Sample Output 2 : Invalid input

ANSWER:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Reflection;



namespace ConsoleApplication2

{

 class Program

{

 static void Main(string[] args)
```

```
{
 string str = Console.ReadLine();
 Console.WriteLine(UserProgramCode.nextString(str));
Console.ReadLine();




      }
}


}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Reflection;
using System.Collections;


namespace nextConsonentorVowel
{
```

```
class UserProgramCode
{
    public static string nextString(string str)
    {
        ArrayList vowel_small = new ArrayList();
        ArrayList vowel_caps = new ArrayList();

        vowel_small.Add('a');
        vowel_small.Add('e');
        vowel_small.Add('i');
        vowel_small.Add('o');
        vowel_small.Add('u');
        vowel_caps.Add('A');
        vowel_caps.Add('E');
        vowel_caps.Add('I');
        vowel_caps.Add('O');
        vowel_caps.Add('U');
        char[] inp = str.ToCharArray();
        char[] out1 = new char[str.Length];

        for (int i = 0; i < str.Length; i++)
        {
            if (!Char.IsLetterOrDigit(str[i]))
```

```
        return "invalidinput";
}
for (int i = 0; i < str.Length; i++)
{

   if (vowel_caps.Contains(inp[i]) || vowel_small.Contains(inp[i]))
   {
      char ch = (char)((int)inp[i] + 1);
      out1[i] = ch;




   }
   else if (inp[i] == 90 || inp[i] == 122)
   {
      if (inp[i] == 90)
         out1[i] = 'A';
      else
         out1[i] = 'a';
   }
   else
   {
      if (inp[i] >= 65 && inp[i] <= 90)
```

```
{
   if (inp[i] > 85)
      out1[i] = 'A';
   else
   {
      foreach (char che in vowel_caps)
      {
         if (che > inp[i])
         {
            out1[i] = che;
            break;
         }
      }
   }
}
else
{
   if (inp[i] > 117)
      out1[i] = 'a';
   else
   {
      foreach (char che in vowel_small)
      {
```

```
                    if (che > inp[i])
                    {
                        out1[i] = che;
                        break;
                    }
                }
            }
        }
    }
    string output = null;
    foreach (char c in out1)
    {


        output += c.ToString();
    }
    return output;


  }
}
```

}

4. Donations

QUESTION:

Donations

Given 2 inputs,string array input1 and integer input2.The usercodes,locations and donations are appended as one element and stored in input1 in the following format, ABCDEFGHI- here the ABC represents the usercode ,DEF represents the location and GHI represents the donation amount. Write a program to find the total amount donated by the users who have the same location code given in input2 integer value. Business rule: 1) If the string array contains any duplicates, then print -1. 2) If the string array contains any special characters, then print -2. Create a class named UserProgramCode that has the following static method

public static int getDonation(string[] input1, int input2)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the string array.

The next 'n' lines of input consists of strings that correspond to elements in the string array.

The next line of the input consists of an integer that corresponds to the location code.

Refer business rules and sample output for output format.

Sample Input 1 :

4 123111241 124222456 145111505 124553567 111

Sample Output 1 : 746 Sample Input 2 : 4 123111241 124222456 124222456 124553567 111

Sample Output 2 : -1

ANSWER:

```
using System;
using System.Text.RegularExpressions;
namespace code1
{
  class Program
{
  static void Main(String[] args)
{
  int n;
Regex reg = new Regex(@"([A-Za-z0-9])$");
n = int.Parse(Console.ReadLine());
  String[] input1 = new String[n];
  int input2;
  int output;
  for (int i = 0; i < n; i++)
{
  input1[i] = Console.ReadLine();
  if (!reg.IsMatch(input1[i]))
{
  Console.WriteLine("-2"); return;
}




          }
  for (int i = 0; i <n; i++)
{
```

```
  for (int j = i+1; j < n; j++)
{
  if(input1[i].Equals(input1[j]))
{   Console.WriteLine("-1");}
}
}
input2 = int.Parse(Console.ReadLine());

output = UserMainCode.getDonation(input1, input2);
  Console.WriteLine(output);
}
}

}
```

```
using System;



public class UserMainCode
{







    public static int getDonation(string[] input1, int input2)
{
String temp;
  int n=input1.Length;
  int output=0,don;;
  for (int i = 0; i < n; i++)
{
  temp = input1[i].Substring(3, 3);
```

```
 if(int.Parse(temp)==input2){
 don=int.Parse(input1[i].Substring(6,3));
 output += don;
}
}
  return output;



  }
}
```

5. Max Diff in Array

QUESTION:

Max Diff in Array

Given an integer input array input,find out the maximum difference between any two elements such that larger element appears after the smaller number in the input array. Business Rules : 1. If any of the given inputs contain any negative number, then print -1. 2. If there is only one element or more than 10 elements in the input array, then print -2. 3. If there are any duplicates in the input array, then print -3.

Create a class named UserProgramCode that has the following static method

public static int diffIntArray(int[] input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.

The next 'n' lines of input consist of elements in the input array.

Output is an integer.

Refer business rules and sample output for formatting specifications.

Sample Input 1 : 7 2 3 10 6 4 8 1

Sample Output 1 :

8

[Hint : (Diff between 2 and 10. 10 is larger than 2)] Sample Input 2 : 5 4 5 -20 9 10

Sample Output 2 :

-1

ANSWER:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace Max_diference

{

   class UserProgramCode

   {

      public static int diffIntArray(int[] input1)

      {

         int diff=0;

         int len =input1.Length;


         int max = 0;

         for (int i = 0; i < input1.Length; i++)
```

```
{
   if (input1[i] < 0)
   {
      return -1;
   }
}
if (len < 2 || len > 10)
   return -2;
for (int i = 0; i < len; i++)
{
   for (int j = i+ 1; j < len; j++)
   {
      if (input1[i] == input1[j])
         return -3;
   }
}


for (int i = 0; i < len; i++)
{
   for (int j = i + 1; j < len; j++)
   {
      if (input1[j] > input1[i])
      {
```

```
                int m = input1[j] - input1[i];
                if (m > max)
                    max = m;
            }
        }
    }
    return max;


  }
 }
}



using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Max_diference
{
  class Program
```

```
  {
    static void Main(string[] args)
    {
      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 op = UserProgramCode.diffIntArray(arr);
      Console.WriteLine(op);
      Console.ReadLine();

    }
  }
}
```

6. List the Elements A

QUESTION:

List the Elements - A

Write a program that accepts integer list and an integer. List all the elements in the list that are smaller than the value of given integer. Print the result in descending order.

Example:

input1: [1,4,7,3,9,15,24]

input2: 17

Output1:[15,9,7,4,3,1]

Include a class UserProgramCode with static method GetElements() which accepts an integer list and the integer (input2) as input and returns an integer list. If there is no element found in input1, then store -1 to the first element of output list. Create a class Program which would get the input and call the static method GetElements() present in the UserProgramCode. If there is no such element in the input list, print "No element found".

Input and Output Format:

Input consists of n+2 integers. The first integer corresponds to n, the number of elements in the array. The next'n' integers correspond to the elements in the array.

The last input is an integer.

Output is an integer list or the string "No element found".

Sample Input 1: 7

1 4

7

3

9

15

24

17

Sample Output 1: 15 9 7 4 3 1 Sample Input 2: 6 5 9 3 4 16 21 9 Sample Output 2: 5 4 3

ANSWER:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace List_the_elements

{

   class Program

   {

      static void Main(string[] args)

      {

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


         List<int> a = new List<int>(n);

         List<int> output = new List<int>();

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

         {


            a.Add(int.Parse(Console.ReadLine()));
```

```
        }
        int chk = int.Parse(Console.ReadLine());
        output= UserProgramCode.GetElements(a,n,chk);
        if (output[0] == -1)
        {

            Console.WriteLine("No Element Is Found");

        }
        if (output[0] != -1)
        {
            for (int i = 0; i < output.Count; i++)
            {

                Console.WriteLine(output[i]);
            }
        }
        Console.ReadLine();
      }
   }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace List_the_elements
{
   class UserProgramCode
   {
      public static List<int> GetElements(List<int> a,int n,int chk)

      {
         List<int> temp = new List<int>();
         int count =0;

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

            if (a[i] < chk)
            {
               temp.Add(a[i]);
               count = 1;
```

```
            }
        }
        temp.Sort();
        temp.Reverse();
        if (count == 0)
        {
            temp.Add(-1);
        }
        return temp;
    }
  }
}
```

7. Is – Is Not

QUESTION:

Is – Is Not

Write a program to read a String and to replace every appearance of the word "is" by "is not". If the word "is" is immediately preceeded or followed by a letter no change should be made to the string. Print the final string.

Example:
input = This is just a misconception
output = This is not just a misconception

Include a class UserProgramCode with a static method negativeString which accepts a string. The return type (String) should return the final output.

Create a Class Program which would be used to accept a string input, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a string.

Output consists of a string.

Refer sample output for formatting specifications.

Sample input 1:

This is just a misconception

Sample Output 1:

This is not just a misconception

ANSWER:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Reflection;


namespace ConsoleApplication2

{

   class Program

   {

      static void Main(string[] args)
```

```
        {
            string s = Console.ReadLine();
            string output = UserProgramCode.negativeString(s);
            Console.WriteLine(output);



        }
    }

}
class UserProgramCode
    {
        public static string negativeString(string str)
        {
            string neg_string = null;
            string[] str1 = str.Split(' ');
            StringBuilder sb = new StringBuilder();
            for (int i=0;i<str1.Length;i++)
            {

                if (str1[i].Equals("is"))
                {
                    sb.Append("is not ");
```

```
            }
            else
            {
                sb.Append(str1[i]+" ");
            }
        }
        neg_string = sb.ToString();
        return neg_string;
    }
}
```

8. convertRomanToDecimal

QUESTION:

Convert Roman to Decimal

Write a program to convert the given roman number to decimal number.

Example : Input string: XVII Output variable:10+5+1+1 = 17
The input string should contain only the alphabets given below (in upper case) . Valid alphabets are I, V, X, L, C, D, and M 'I': The corresponding value = 1 'V': The corresponding value = 5 'X': The corresponding value =10 'L': The corresponding value = 50 'C': The corresponding value = 100 'D': The corresponding value = 500 'M': The corresponding value =1000

Include a class UserProgramCode with a static method convertRomanToDecimal that accepts a string

and returns an integer. The method returns -1 if the input string is not valid.

Create a Class Program which would be used to read the string and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a string.

Output consists of an integer.

Sample Input 1: XII Sample Output 1: 12 Sample Input 2: DCL Sample Output 2: 650

Sample Input 3: DCA Sample Output 3: -1

ANSWER:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace Roman_to_decimal

{

   class UserProgramCode

   {

      public static int RomanToDecimal(string str)

      {

         List<string> romanList = new List<string>();

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

```
int sum = 0;


romanList.Add("I");

romanList.Add("V");

romanList.Add("X");

romanList.Add("L");

romanList.Add("C");

romanList.Add("D");

romanList.Add("M");


intList.Add(1);

intList.Add(5);

intList.Add(10);

intList.Add(50);

intList.Add(100);

intList.Add(500);

intList.Add(1000);


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

{

   if (romanList.Contains(str[i].ToString()))

      sum = sum + intList[romanList.IndexOf(str[i].ToString())];

   else
```

```
                return -1;
            }
            return sum;

        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Roman_to_decimal
{
    class Program
    {
        static void Main(string[] args)
        {
            string str = Console.ReadLine();
            int op = UserProgramCode.RomanToDecimal(str);
            Console.WriteLine(op);
            Console.ReadLine();
        }
```

```
    }
}
```

9. Count of Elements

QUESTION:

113013

Count of Elements

Write a program that gets the count of elements in input1 list that starts with the character passed in input2 irrespective of case. Print the count.

Example:

input1: ['abc','Apple','Mango']

input2: a

Output1:

2

Business Rule:

1. If there is no element that start with the given char in input1, then return -1.
2. Only alphabets should be given in input1 string else return -2.

Include a class UserProgramCode with a static method GetCount which accepts the size of the string array, string array and a character. The return type (Integer) should return count. Follow the Business rules.

Create a Class Program which would be used to accept the size of the array, the array elements and a character, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of an integer, which corresponds to the size of the array, a string list, and a character.

Output consists of an Integer(final count), or a String("No elements Found" if -1 is returned or "Only alphabets should be given" if -2 is returned.

Refer sample output for formatting specifications.

Sample Input 1:

3

abc

Apple

Mango

a

Sample Output 1:

2

Sample Input 2:

2

goods

bads

a

Sample Output 2:

No elements Found

Sample Input 3:

2

good$

bad$

a

Sample Output 3:

Only alphabets should be given

ANSWER:

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Reflection;

namespace ConsoleApplication2
{
   class Program
   {
      static void Main(string[] args)
      {
         int n = int.Parse(Console.ReadLine());
         string[] str = new string[n];
         if (n > 0)
         {
            for (int i = 0; i < n; i++)
            {
               str[i] = Console.ReadLine();
            }
            char c = char.Parse(Console.ReadLine());
            int output = UserProgramCode.getCount(n, str, c);
            if (output > 0)
            {
```

```
            Console.WriteLine(output);
        }
        else if (output == -1)
        {
            Console.WriteLine("No elements Found");
        }
        else if (output == -2)
        {
            Console.WriteLine("Only alphabets should be given");
        }
      }


    }
  }

}

class UserProgramCode
  {
    public static int getCount(int size,string[] str,char c)
    {
      int count=0;
```

```
char c_cap = c;
char c_small = c;
Regex reg = new Regex(@"^([A-Za-z]{1,})$");
foreach (string s in str)
{

   if (!reg.IsMatch(s))
      return -2;
   string ch = c.ToString();
   if (c >= 97 && c <= 122)
   {
      c_cap = (char)((int)(c) - 32);

   }
   else if (c >= 65 && c <= 90)
   {
      c_small = (char)((int)(c) + 32);
   }

   char[] inp = s.ToCharArray();


   if (c_small == inp[0]||c_cap==inp[0])
```

```
            count++;


        }
        if (count >= 1)
        {
           return count;
        }
        else
           return -1;
     }
  }
```

10. Digit Sum in String Array

QUESTION:

Digit Sum in String Array

Given a String array as input. Each element in this array may contain alphabets or digits. Develop code to add all the digits in every string and print the sum as an int. If two digits appear simultaneously do not consider it as one number. Ex- For 'Hyderabad 21' consider 2 and 1 as two digits instead of 21 as a number. Business Rules : 1. If there are any special characters in the input strings, then print -1. Example 1: input1: 5 input2: AAA1 B2B 4CCC A5 ABCDE output1:1+2+4+5=12 Example 2: input1 : 3 input2 : 12 C23 5CR2 output1 : 1+2+2+3+5+2 = 15

Create a class named UserProgramCode that has the following static method

public static int sumOfDigits(string[] input1)

Create a class named Program that accepts the inputs and calls the static method present in the

UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input string array.

The next 'n' lines of input consist of elements in the input string array.

Refer sample output for formatting specifications.

Sample Input : 5 AAA1 B2B 4CCC A5 ABCDE

Sample Output :

12

ANSWER:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;




namespace SumOfdigitinString

{

   class Program

   {

      static void Main(string[] args)
```

```
{
    string s = Console.ReadLine();
    int result = UserProgramCode.sumofdigitsinstring(s);
    Console.WriteLine(result);
    Console.ReadLine();
}
public static int sumofdigitsinstring(string input1)
{
    int temp, sum = 0;
    StringBuilder sb = new StringBuilder();
    // string input1 = "app123e";
    char[] ch = input1.ToCharArray();
    foreach (char c in ch)
    {
        if (char.IsDigit(c))
        {
            sb.Append(c);
        }

    }
    string s = sb.ToString();
    int n = int.Parse(s);
    while (n > 0)
```

```
        {
            temp = n % 10;
            sum = sum + temp;
            n = n / 10;
        }
        return sum;
    }
  }
}
```

11. Calculate the Efficiency

QUESTION:

Calculate the Efficiency

In a company, worker efficiency is determined on the basis of the time required for a worker to complete a particular job.If the time taken by the worker is input, then display the efficiency of the worker.

If time taken is 1 to 3 hours then the worker is said to be "Highly efficient".

If time taken is more than 3 hrs and less than or equal to 4 hours then efficiency level is "Improve speed"

If time taken is more than 4 hours and less than or equal to 5 hours then efficiency level is "Need Training" to improve speed.

If the time taken is more than 5 hours, then the worker has to "Leave the company".

otherwise it should return Invalid Input

Include a class UserProgramCode with static method EfficiencyChecker which accepts float.The return type is String.

Create a class Program which would get the input and call the static method EfficiencyChecker present in the UserProgramCode.

Input output format The input consists a float. The output consists the String. Sample Input 1: 5.0 Sample Output 1: Need Training Sample Input 2: 10.5 Sample Output 2: Leave the company Sample Input 2: -2 Sample Output 2: Invalid Input

ANSWER:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace progm51

{

   class Program

   {

      static void Main(string[] args)

      {

         decimal inp=Convert.ToDecimal(Console.ReadLine());

         string outp=UserProgramCode.EfficiencyChecker(inp);

         Console.WriteLine(outp);

      }

   }
```

```
class UserProgramCode
  {
     public static string EfficiencyChecker(decimal a)
     {
        if ((a >= 1) && (a <= 3))
        {
           return "Highly efficient";
        }
        else if ((a > 3) && (a <= 4))
        {


           return "Improve speed";
        }
        else if ((a > 4) && (a <= 5))
        {
           return "Need Training";


        }
        else if (a > 5)
        {
           return "Leave the company";
```

```
        }
        else
        {
            return "Invalid Input";
        }
      }

  }
}
```

12. Sum of Cube

QUESTION:

Calculate the sum of cube

Write a program to find the sum of the cube of first 'n' natural numbers.

Example:
input = 5
output = 225

Include a class UserProgramCode with a static method sumOfCube() that accepts an integer and returns an integer . If the input is not a natural number, return -1. Create a class Program which would get the

input and call the static method sumOfCube() present in the UserProgramCode.

Input and Output Format:

Input is an integer that corresponds to n

Output is an integer (Sum of cubes) or if the given input n is not a natural number then print “The input is not a natural number”

Sample Input 1:

5

Sample Output 1:

225

Sample Input 2:

-1

Sample Output 2:

The input is not a natural number

ANSWER:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace ConsoleApplication2

{

   class Program
```

```
  {
    static void Main(string[] args)
    {

      {
        int n = int.Parse(Console.ReadLine());
        int sl = UserProgramCode.sumOfCube(n);
        if (sl == -1)
          Console.WriteLine("The input is not a natural number");
        else
          Console.WriteLine(sl);
        Console.ReadLine();

      }
    }
  }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
```

```
using System.Text;


namespace ConsoleApplication2
{
   class UserProgramCode
   {
      public static int sumOfCube(int n)
      {
         if (n < 0)
            return -1;
         int sum = 0;
         for (int i = 1; i <= n; i++)
            sum = sum + (i * i * i);
         return sum;


      }
   }
}
```

13. Vehicle Mileage

QUESTION:

Vehicle Mileage

Write a program to find the mileage of the vehicle given the bike caliber (input1) and the number of years (input2) the vehicle is being used as inputs and print the output as given in the sample output. The

ideal mileage details given by the manufacturer are as follows : if CC is between 100 to 125 mileage is 75 if CC is between 126 to 135 mileage is 70 if CC is between 136 to 150 mileage is 60 if CC is between 151 to 200 mileage is 50 if CC is between 201 to 220 mileage is 35 But with the years of usage of the vehicle, the promised mileage is reduced as follows : 1.Vehicle Usage till 2 years, the mileage is reduced by 10% than the promised. 2.Vehicle Usage more than 2 years till 4 years, the mileage is reduced by 15% than the promised. 3.Vehicle Usage more than 4 years, the mileage is reduced by 20% than the promised. The output to be printed is

The mileage of the bike is xyz

where xyz is the calculated mileage. Business Rules :- 1) If the bike caliber given is negative number or below 100, then print 'Invalid caliber'. 2) If the number of years given is a negative number or more than 20 years, then print 'Invalid years'. Create a class named UserProgramCode that has the following static method

public static double findMileage(int input1, int input2)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of 2 integers.

The first line of the input corresponds to bike calibre and the second line of the input corresponds to number of years.

Refer sample output and business rule for output formatting specifications.

Sample Input 1 : 100 1

Sample Output 1 :

The mileage of the bike is 67.5 Sample Input 2 : 90 1

Sample Output 2 :

Invalid caliber

ANSWER:

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Vechile
{
   class Program
   {
      static void Main(string[] args)
      {
         int cc = int.Parse(Console.ReadLine());
         int year = int.Parse(Console.ReadLine());
         double op = UserProgramCode.findmileage(cc, year);
         if (op == -1)
         {
            Console.WriteLine("Invalid Caliber");
         }
         else
            Console.WriteLine("The mileage of the bike is {0}",op);
         Console.ReadLine();
      }
   }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace Vechile
{
   class UserProgramCode
   {
      public static double findmileage(int cc, int year)
      {
         double mileage = 0;
         if (cc >= 100 && cc <= 125)
         {
            if (year < 2)
            {
               mileage = 75 - (75 * 0.1);
            }
            if (year > 2 && year <= 4)
            {
```

```
        mileage = 75 - (75 * 0.15);
    }
    if (year > 4)
    {
        mileage = 75 - (75 * 0.20);
    }
}
else if (cc >= 126 && cc <= 135)
{
    if (year < 2)
    {
        mileage = 70 - (70 * 0.1);
    }
    if (year > 2 && year <= 4)
    {
        mileage = 70 - (70 * 0.15);
    }
    if (year > 4)
    {
        mileage = 70 - (70 * 0.20);
    }
}
else if (cc >= 136 && cc <= 150)
```

```
{
   if (year < 2)
   {
      mileage = 60 - (60 * 0.1);
   }
   if (year > 2 && year <= 4)
   {
      mileage = 60 - (60 * 0.15);
   }
   if (year > 4)
   {
      mileage = 60 - (60 * 0.20);
   }
}
else  if (cc >= 151 && cc <= 200)
{
   if (year < 2)
   {
      mileage = 50 - (50 * 0.1);
   }
   if (year > 2 && year <= 4)
   {
      mileage = 50 - (50 * 0.15);
```

```
    }
    if (year > 4)
    {
        mileage = 50 - (50 * 0.20);
    }
}
else if (cc >= 201 && cc <= 220)
{
    if (year < 2)
    {
        mileage = 35 - (35 * 0.1);
    }
    if (year > 2 && year <= 4)
    {
        mileage = 35 - (35 * 0.15);
    }
    if (year > 4)
    {
        mileage = 35 - (35 * 0.20);
    }
}
else
    return -1;
```

```
        return mileage;
    }
}
```

14. Get Common Elements

QUESTION:

Get Common Elements

Write a program that accepts two lists and finds out the elements that are common in both of the given input lists and consolidates the common elements in another list. Business Rule: Only positive number should be given in input List. Else print -1 in the output. Create a class named UserProgramCode that has the following static method

public static List<int> getCommonItems(List<int> input1, List<int> input2)

Create a class named Program that accepts the 2 input lists and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer that corresponds to m, the number of elements in the first list. The next 'm' lines consist of elements in the first list.

The next line of the input consists of an integer that corresponds to n, the number of elements in the second list. The next 'n' lines consist of elements in the second list.

Output consists of the list that contains the common elements in both the lists. If any of input list elements are negative, print -1.

Sample Input : 5 5 6 7 8 1 4 5 2 3 1

Sample Output : 5 1

ANSWER:

```
-----Get Common Elements----
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace getcommonelements
{
   class Program
   {
      static void Main(string[] args)
      {
         int m, n,i,k,c;
         List<int> listm=new List<int>();
         List<int>listn=new List<int>();
         List<int> listo = new List<int>();
         List<int> final = new List<int>();
         m=Convert.ToInt32(Console.ReadLine());
         for(i=0;i<m;i++)
         {
            k=Convert.ToInt32(Console.ReadLine());
```

```
            listm.Add(k);
        }
         n=Convert.ToInt32(Console.ReadLine());
        for(i=0;i<n;i++)
        {
            c=Convert.ToInt32(Console.ReadLine());
            listn.Add(c);
        }
        listo = UserProgramCode.getCommonItems(listm, listn);
        final = listo.Distinct().ToList();
        foreach (int t in final)
        {
            Console.WriteLine(t);
        }
        Console.ReadLine();
      }
   }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
namespace getcommonelements
{
   class UserProgramCode
   {
      public static List<int> getCommonItems(List<int> input1, List<int> input2)
      {
         int len1, len2;
         List<int> p = new List<int>();
         len1 = input1.Count;
         len2 = input2.Count;
         foreach (int i in input1)
         {
            foreach (int k in input2)
            {
               if (i == k)
               {
                  p.Add(i);
               }
            }
         }
         return p;
      }
```

```
    }
}
```

15. Find largest digit in a given number

QUESTION:

Find largest digit in a given number
Write a code to find the Largest digit from given input integer.
Include a class UserProgramCode with static method findLargestDigit(int num)
Create a class Program which would get the input and call the static method findLargestDigit(num) present in the UserProgramCode.
If the interger is a negative value findLargestDigit(num) method returns -1 to Program, otherwise returns largest digit in a given number. If -1 is returned then print "The number should be a positive number". Input and Output Format: Input is an integer n. Output is an integer which is the largest digit in the given number n.
SAMPLE INPUT 1:
456
SAMPLE OUTPUT1:
6 SAMPLE INPUT 2: -12434567 SAMPLE OUTPUT2: The number should be a positive number

ANSWER:

19)

```
using System;
using System.Collections.Generic;
using System.Linq;
```

```
using System.Text;


namespace Fwd_Prgs
{
    public class UserProgramCode
    {
        public static int findLargestDigit(int num)
        {
            if (num > 0)
            {
                int temp, res = 0;


                while (num > 0)
                {
                    temp = num % 10;
                    if (temp > res)
                        res = temp;
                    num = num / 10;
                }
                return res;
            }
            else
                return -1;
```

```
    }
  }

  class Program
  {
    static void Main(string[] args)
    {
      int n = int.Parse(Console.ReadLine());
      int res = UserProgramCode.findLargestDigit(n);
      if(res!=-1)
        Console.WriteLine(res);
      else
        Console.WriteLine("The number should be a positive number");

    }
  }
}
```

16. Employee Designation

QUESTION:

Employee Designation

Given an input1 string array in the format {Employee1, Designation, Employee2, Designation, Employee3, Designation, and so on... } and a string input2, write a program to fetch the employee

names from input1 based on input2 (designation) value and assign it in an output array and print the array. Case sensitivity can be ignored. Business rule: 1) If input1 or input2 contains any special characters, then print 'Invalid Input' 2) If input1 does not contain the designation in input2, then print 'No employee for ABC designation' where ABC is the Input2 value. 3) If all the employees belong to the same designation, then print 'All employees belong to same ABC designation' where ABC is the Input2 value. Example 1: input1: Ram Manager Ganesh Developer Srijith Developer input2: Developer output : Ganesh Srijith Example 2: Input 1: Manish BiDeveloper Babu Manager Rohit Associate Input 2: System Analyst Output1: No employee for System Analyst designation Create a class named UserProgramCode that has the following static method

public static string[] getEmployee(string[] input1, string input2)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the string array.

The next 'n' lines of input consists of strings that correspond to elements in the string array.

The next line of the input consists of a string that corresponds to the Designation.

Refer business rules and sample output for output format.

Sample Input 1:

6 Ram Manager Ganesh Developer Srijith Developer Developer

Sample Output 1:

Ganesh Srijith

Sample Input 2:

6 Manish BiDeveloper Babu Manager Rohit Associate System Analyst

Sample Output 2: No employee for System Analyst designation

ANSWER:

Employee designation

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace EmployeeDesgination
{
   class Program
   {
      static void Main(string[] args)
      {
         int n = int.Parse(Console.ReadLine());
         string[] str = new string[n];
         for (int i = 0; i < n; i++)
         {
            str[i] = Console.ReadLine();
         }
         string find = Console.ReadLine();
         string[] op = UserProgramCode.getEmployee(str, find);
```

```
        if (op.Length == str.Length / 2)
        {
           Console.WriteLine();
        }
        if (op.Length == 0)
        {
           Console.WriteLine();
        }

        foreach (string item in op)
        {
           Console.WriteLine(item);
        }
        Console.ReadLine();
      }
   }


}
```

userprogram

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace EmployeeDesgination
{
   class UserProgramCode
   {
      public static string[] getEmployee(string[] input1, string input2)
      {
         List<string> list = new List<string>();
         for (int i = 0; i < input1.Length; i++)
         {
            if (input2.ToLower() == input1[i].ToLower())
            {
               list.Add(input1[i-1]);
            }
         }
         return list.ToArray();
      }
   }
}
```

17. Replace String

QUESTION:

Replace String

Write a program to form a new string by replacingeach character in the n'th word of the input string with given special character. Display resultant string in lowercase.

Example:

Input1: Hi are you fine Ram
Input2: 5
Input3: *
Output:hi are you fine ***

Business Rules:

1. Only alphabets should be given in input1 string Else return "-1" from the method and print “Invalid String” in Main.
2. Only positive number should be given for input2 Else return "-2" from the method and print “Number not positive” in Main.
3. Only special characters should be given for input3 Else return "-3" from the method and print “Character not a special character” in Main.

Include a class UserProgramCode with a static method replaceString which accept a String, an integer and a character. The return type (String) should return the Final String.
Create a Class Program which would be used to accept a String, an integer and a character , and call the static method present in UserProgramCode.
Input and Output Format:
Input consists of a String, an integer and a character, where String corresponds to the input string, the integer corresponds to the word number and the character values corresponds to the change character.

Output consists of a String.

Refer sample output for formatting specifications.

Sample Input 1:

Hi are you fine Ram

5

*

Sample Output 1:

hi are you fine ***

Sample Input 2:

Hi @re you fine Ram

5

*

Sample Output 2:

Invalid String

Sample Input 3:

Hi are you fine Ram

-5

*

Sample Output 3:

Number not positive

Sample Input 4:

Hi are you fine Ram

5

o

Sample Output 4:

Character not a special character

ANSWER:

REPLACE STRING

```
using System;



class Program
{
  public static void Main( string[] args )
  {
        string inputWord=Console.ReadLine();
        int position=Convert.ToInt32(Console.ReadLine());
        char ch=Convert.ToChar(Console.ReadLine());
        string result=UserProgramCode.replaceString(inputWord,position,ch);
        if(result.Equals("-1"))
              Console.WriteLine("Invalid String");
        else if(result.Equals("-2"))
              Console.WriteLine("Number not positive");
        else if(result.Equals("-3"))
              Console.WriteLine("Character not a special character");
        else
              Console.WriteLine(result);
```

```
 }
}
using System;


class UserProgramCode
{
   public static string replaceString(string inputWord, int position, char ch)
   {
      string inputWord1=inputWord.ToLower();
      foreach (Char z in inputWord)
      {
         if (!(Char.IsLetterOrDigit(z) || Char.IsWhiteSpace(z)))
         {
            return "-1";
         }
      }
      if (position <= 0)
      {
         return "-2";
      }
      if ((Char.IsLetterOrDigit(ch)) || Char.IsWhiteSpace(ch))
      {
         return "-3";
```

```
        }
        else
        {
            string[] A = inputWord1.Split(' ');
            string b = string.Copy(A[position - 1]);
            char[] B = b.ToCharArray();
            for (int i = 0; i < b.Length; i++)
            {
                B[i] = ch;
            }
            string c = new string(B);
            A[position - 1] = c;
            string d = string.Join(" ", A);
            return d;
        }
    }
}
```

18. Form String

QUESTION:

Form String

Given a String array and an int 'n', write a program to perform the following operations: 1) Pick nth character from each String element in the String array and form a new String. 2) If nth character not

available in a particular String in the array consider $ as the character. 3) Print the new String. Business Rules : 1. If there are any special characters in the input strings, then print -1.

Create a class named UserProgramCode that has the following static method

public static string formString(string[] input1,int input2)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer 'k' that corresponds to the number of elements in the string array.

The next 'k' lines of the input consists of strings that correspond to the elements in the string array.

The next line of the input consists of an integer that corresponds to n.

Refer sample output for formatting specifications.

Sample Input : 4 ABC XYZ EFG MN 3 Sample Output : CZG$

ANSWER:

1.Form string

FIRST TRY THIS

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace formstr78
```

```
{
   class Program
   {
      static void Main(string[] args)
      {
         int n = Convert.ToInt32(Console.ReadLine());
         string[] s = new string[n];
         for (int i = 0; i < s.Length; i++)
         {
            s[i] =Console.ReadLine();
         }
         int p = Convert.ToInt32(Console.ReadLine());
      string z=   form.fun(s, p);
      if (z == "-2")

         Console.WriteLine("-2");
      else
         Console.WriteLine(z);
      Console.ReadLine();
      }
   }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;

namespace formstr78
{
   class form
   {
      public static string fun(string[] a, int f)
      {
         //StringBuilder sb = new StringBuilder();
         StringBuilder sb1 = new StringBuilder();
         string p = @"[a-zA-Z]+$";
         for (int i = 0; i < a.Length; i++)
         {
            if (!Regex.IsMatch(a[i], p))
               return "-2";
            int len = a[i].Length;
            if (len < f)
               sb1.Append("$");
```

```
            else


                sb1.Append(a[i].Substring(len-1));
            }
            string z = sb1.ToString();
            return z;




        }
    }
}
```

2.

```
//program.cs

using System;

using System.Collections.Generic;

using System.Linq;
```

```
using System.Text;
using System.Collections;
namespace trial
{
   class Program
   {
      static void Main(string[] args)
      {
         String str = Console.ReadLine();
         String res = UserProgramCode.getSpecialChar(str);
         Console.WriteLine(res);
      }
   }
}

//UserProgramCode.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace trial
{
```

```
class UserProgramCode
{
    public static string getSpecialChar(string input1)
    {
        int num = 0;
        StringBuilder sb=new StringBuilder();
        int sp = 0;
        int len = input1.Length;
        for (int i = 0; i < len; i++)
        {
            char c = input1[i];
            if (char.IsDigit(c))
            {
                num++;
                sb.Append(c);
            }
            if (c == '#' || c == '$' || c == '%' || c == '&')
            {
                sp++;
                sb.Append(c);
            }
            //else if (!char.IsLetter(c))
            //{
```

```
            //   sb.Append(c);
            //}
        }
        if (num == 0 || sp == 0)
        {
            return "-1";
        }
        else
            return sb.ToString();

    }
  }
}
```

19. Calculate Commission

QUESTION:

Calculate Commission Write a program to calculate the commission on given sales as per the following policy. Include a class UserProgramCode with a static method calculateCommission which accepts a float as input.

Create a class Program which would get the input and call the static method calculateCommission present in the UserProgramCode. If the method returns -1, then print 'Invalid Input'. If sales is less than Rs. 10000/- no commission. If sales is between Rs. 10000/- to Rs. 25000/- commission is 10% of sales. If sales is more than Rs. 25000/- then commission is Rs. 500/- plus 8% of sales amount. Business Rule : 1. If input is negative number then the method calculateCommission returns -1. 2. Otherwise return a

calculated commission. Input and Output format : Input consists of float. Refer sample output for formatting specifications. Sample Input 1 : 11000 Sample Output 1: 1100 Sample Input 2 : -1000 Sample Output 2 : Invalid Input

ANSWER:

70) using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication26

{

class Program

{

static void Main(string[] args)

{

float s = float.Parse((Console.ReadLine()));

int a = UserProgramCode.calculateCommission(s);

if (a == -1)

Console.WriteLine("Invalid Input");

else

```
            Console.WriteLine(a);
        Console.ReadLine();




    }
}
class UserProgramCode
{
    public static int calculateCommission(float a)
    {
        double com;
        double r = Convert.ToDouble(a);


        if (r < 0)
        {
            return -1;
        }
        else if (r < 10000)
        {
            return 0;
        }
        else if (r >= 10000 && r <= 25000)
        {
```

```
            com = r * 1.0 / 10.0;
            return Convert.ToInt32(com);



        }

        else if (r > 25000)
        {
            com = 500 + (r * 8.0/ 100.0);
            return Convert.ToInt32(com);

        }
        return -1;

    }
  }
}
```

20. Count Subsets

QUESTION:

Count Subsets

Given a method with an integer list as input, Write code to find the number of subsets formed from the

given input. Consider any three elements for the input list which forms as one subset. Sum of first two elements must be equal to third element. The number of subsets which satisfy these conditions would be the output that is printed Note: The elements in a subset should satisfy below conditions: 1)Any subset should have only 3 elements 2)The elements in each subset must be distinct Business rule: 1) Print -1 when no such subsets are formed 2) Print -2 if input list consists of negative elements 3) Print -3 when same integer element is repeated twice in the input list. Example: input 5 Input1 1 2 3 4 6 The subsets formed are (1,2,3), (1,3,4), (2,4,6) output = 3 Create a class named UserProgramCode that has the following static method

public static int countSubsets(int[] input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.

The next 'n' lines of input consist of elements in the input array.

Output consists of an integer.

Refer business rules and sample output for formatting specifications.

Sample Input 1 : 5 1 2 3 4 6

Sample Output 2 :

3

ANSWER:

52.CountSubsets

```
using System;

  class Program

  {

    public static void Main( string[] args )

    {
```

```
                int arrSize,output;
                arrSize = Convert.ToInt32(Console.ReadLine());
                int[] input = new int[arrSize];
                for(int i=0;i<arrSize;i++)
                {
                        input[i] = Convert.ToInt32(Console.ReadLine());
                }
                output = UserProgramCode.countSubsets(input);
                Console.WriteLine(output);
      Console.Read();
    }
  }
using System;
class UserProgramCode {
          public static int countSubsets(int[] input1)
           {
       int count = 0,flag=0;
       foreach (int l in input1)
       {
          if (l < 0)
          {
             flag++;
             return -2;
```

```
    }
}
for (int a = 0; a < input1.Length; a++)
{for(int b=a+1;b<input1.Length;b++)
    if (input1[a] == input1[b])
    {
        flag++;
        return -3;
    }
}
if (flag == 0)
{
    for (int i = 0; i < input1.Length; i++)
    {
        for (int j = i + 1; j < input1.Length; j++)
        {
            for (int k = j + 1; k < input1.Length; k++)
            {
                if (input1[i] + input1[j] == input1[k])
                    count++;
            }
        }
    }
```

```
        }
        if (count == 0)
           return -1;
        else
           return count;
            }
}
```

21. Duplicate Characters

QUESTION:

Duplicate Characters

Write a Program which removes duplicate characters from the string. Your program should read a sentence (string) as input from user and return a string removing duplicate characters. Retain the first occurance of the duplicate character. Assume the characters are case – sensitive.

Include a class UserProgramCode with a static method removeDuplicates which accepts a string and returns a string.

Create a Class Program which would be used to accept the input string and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a string with maximum size of 100 characters.

Output consists of a single string.

Refer sample output for formatting specifications.

Sample Input 1:

hi this is sample test

Sample Output 1:

hi tsample

Sample Input 2:

ABC DEF

Sample Output 2:

ABC DEF

ANSWER:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace duplicatecharacters

{

   class Program

   {

      static void Main(string[] args)

      {

         string input;

         input = Console.ReadLine();

         Class1 c=new Class1();
```

```
            string output=c.removeDuplicates(input);
            foreach (char ch in output)
            {
               if (ch != '*')
               {
                  Console.Write(ch);
               }
            }
            //Console.WriteLine(s);
            Console.ReadLine();
        }
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace duplicatecharacters
{
    class Class1
    {
```

```
public string removeDuplicates(string input)
{
   //string[] s=new string[10];
   //s=input.Split(
   int i,j;
   char[] c = new char[100];
   c = input.ToCharArray();
  int length= c.Length;
  for (i = 0; i < length; i++)
  {
     for (j = i + 1; j < length; j++)
     {
        if (c[i] == c[j])
        {
           c[j] = '*';
        }
     }

  }
  string output = new string(c);
  return output;
}
```

```
    }
}
```

22. Digit Sum

QUESTION:

Digit Sum

Given a non-negative int n, return the sum of its digits. If sum is greater than 9 repeat the process and calculate the sum once again until the final sum comes to single digit. Example 1: Input=9999 Output: 9 (9+9+9+9 = 36 and 3+6 = 9) Example 2: Input=698 Output: 5 (6+9+8 = 23 and 2+3 = 5)

Create a class named UserProgramCode that has the following static method

public static int getDigitSum(int input1)

Create a class named Program that accepts the input and calls the static method present in the UserProgramCode.

Input and Output Format: Input consists of an integer.

Output consists of an integer.

Sample Input :

9999

Sample Output :

9

ANSWER:

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;


class UserProgramCode
{
    public static int getDigitSum(int input)
    {
        int sum = 0;
        int sum1 = 0;
        int i = 0;
        int j = 0;
        while (input > 0)
        {
            i = (input % 10);
            sum = sum + i;
            input = input / 10;
        }
        if (sum < 10)
        {
            return sum;
        }
        else
```

```
        {
            while (sum > 0)
            {
                j = (sum % 10);
                sum1 = sum1 + j;
                sum = sum / 10;
            }
        }
        return sum1;
    }
}
class Program
{
    public static void Main(string[] args)
    {
        int input, output;
        input = Convert.ToInt32(Console.ReadLine());
        output = UserProgramCode.getDigitSum(input);
        Console.WriteLine(output);
        Console.ReadLine();
    }
}
```

23. EMI Calculation

QUESTION:

EMI Calculation

A person wants to apply for loan, but bank wants to check whether the person is current employee or retired or student based on given Date Of Birth (input1) and Loan EMI (input2). 1) The Bank can approve loan for student of Rs 200000 with rate of interest of 3% per Annum if age <= 22. 2) The Bank can approve loan for employee of Rs 300000 with rate of interest of 5% per Annum if age>22 and age=<45. 3) The Bank can approve loan for retired of Rs 500000 with rate of interest of 7% per Annum if age>45 and age<=100. Write a program to calculate the EMI . Note :

i) The Loan Amount has to be cleared in either 12/24/36/48 EMI months which is provided as input2. ii) Round the output to the nearest integer if required. iii)Age calculation to be done with respect to current date. Business Rules: i) If Date of Birth is not given in proper fomat(dd-MM-yyyy), then print -1. ii) If EMI month is not 12 , 24 , 36 or 48 , then print -2.

Create a class named UserProgramCode that has the following static method

public static int checkEmpAgeEligible(String input1, int input2)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of a string that corresponds to Date of Birth and an integer that corresponds to EMI months.

Refer business rules and sample output for formatting specifications.

Sample Input 1 :

01-11-1983 12

Sample Output 1 :

26250 Sample Input 2 : 01/11/1983 48

Sample Output 2 :

-1 Sample Input 3 :

01-11-1983 48

Sample Output 3 :

7500

ANSWER:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;



namespace EMI

{

   class Program

   {

      static void Main(string[] args)

      {

         string dob = Console.ReadLine();

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

         int op = UserProgramCode.inst(dob, years);

         {

            Console.WriteLine(op);

         }

      }

   }

}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Xml;
using System.Text.RegularExpressions;
namespace EMI
{
   class UserProgramCode
   {
      public static int inst(string dob, int month)
      {
         if ((month != 12) && (month != 24) && (month != 36) && (month != 48))
         {
            return -2;
         }
```

```
        DateTime dt1;
        int inst = 0;
        double amount1;
        bool i = DateTime.TryParseExact(dob, "dd-mm-yyyy", null,
System.Globalization.DateTimeStyles.None, out dt1);
        if (i == true)
        {
           int year = DateTime.Now.Year - dt1.Year;
           int mont = DateTime.Now.Month - dt1.Month;
           string maxdate = "2022-06-06";
           DateTime dt2 = Convert.ToDateTime(maxdate);
           DateTime dt3 = DateTime.Now.Date;
           DateTime dt4 = dt3.AddMonths(month);
           if (dt4 > dt2)
           {
              return -4;
           }

           if (mont < 0)
           {
              year = year - 1;
              mont = mont + 12;
           }
```

```
            if (year <= 22)
            {
               amount1 = (double)200000 * 1.03;
               inst = (int)amount1 / month;
            }
            if (year > 22 && year <= 45)
            {
               amount1 = (double)300000 * 1.05;
               inst = (int)amount1 / month;
            }
            if (year > 45 && year <= 100)
            {
               amount1 = (double)500000 * 1.07;
               inst = (int)amount1 / month;
            }
            return inst;
         }
         else
            return -1;
      }
   }
}
```

24. validatePassword

QUESTION:

Validate Password Write a method to validate given password. Apply following validations: 1. Minimum length should be 8 characters 2. Must contain any one of these three special characters @ or _ or # 3. May contain numbers or alphabets. 4. Should not start with special character or number 5. Should not end with special character Include a UserProgramCode with a static method validatePassword.The method must return an integer 1 or -1. if it returns 1 then print a message "Valid Password". If the method returns -1 then print a message "Invalid Password". Create a class Program which gets a string as input and calls the static method validatePassword present in the UserProgramCode. Input and Output Format: Input is a string which is the password. Output is also a string which prints a message "Valid Password" or "Invalid Password". Sample Input 1: #bzdfh123c Sample Output 1: Invalid Password Sample Input 2 : jgu_123dfsd3 Sample Output 2: Valid Password

ANSWER:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


class Program

{

  public static void Main( string[] args )
```

```
 {
        String password=Console.ReadLine();
   int result=UserProgramCode.validatePassword(password);
        if(result==1)
            Console.WriteLine("Valid password");
        else
            Console.WriteLine("Invalid password");
   Console.ReadKey();
 }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

class UserProgramCode
{
  public static int validatePassword(String input)
  {
     int output1 = 0;
     if (input.Length >= 8)
     {
        char[] ch = input.ToCharArray();
```

```
        if (char.IsLetter(ch.ElementAt(0)) && char.IsLetterOrDigit(ch.ElementAt(ch.Length - 1)))
        {
            int splchar = 0;
            for (int i = 0; i < ch.Length - 2; i++)
            {
                if (char.IsLetterOrDigit(ch.ElementAt(i)))
                {


                }
                else if (ch.ElementAt(i) == '#' || ch.ElementAt(i) == '_' || ch.ElementAt(i) == '@')
                    splchar++;
            }
            if (splchar >= 1)
            {
                output1 = 1;
            }
        }
        else
            output1 = -1;
    }
    return output1;
  }
}
```

25. Reverse Number

QUESTION:

Reverse Number

Write a program to read a positive number as input and to get the reverse of the given number and print it as output.

Example:

input = 543

output = 345

Include a class UserProgramCode with a static method reverseNumber which accepts an Integer. The return type (Integer) should return the reverse of the given input.

Create a Class Program which would be used to accept an Integer, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of an Integer.

Output consists of an Integer, the reverse of the given Input.

Refer sample output for formatting specifications.

Sample Input 1:

543

Sample Output 1:

345

ANSWER:

question 39 :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace question36
{
   class Program
   {
      static void Main(string[] args)
      {
         int n, c=0;
         n = Convert.ToInt32(Console.ReadLine());
         if (n > 0)
         {
            c = UserProgramCode.reverseNumber(n);
         }
         Console.WriteLine(c);
      }
```

```
}



class UserProgramCode
{
   public static int reverseNumber(int a)
   {
      int rem, sum = 0;

      while (a > 0)
      {

         rem = a % 10;

          sum = (sum*10) + rem;
         a = a / 10;
      }

      return (sum);

   }

}
```

}

26. Reverse and Format

QUESTION:

Reverse and Format

Write a program to read a String and a character and to reverse the string and return it in a format such that each character is separated by the given character. Print the final string.

Example:

input1: "Rabbit"

input2: '-'

output: "t-i-b-b-a-R"

Include a class UserProgramCode with a static method reshape which accepts a string and a character. The return type (String) should return the final String.

Create a Class Program which would be used to read a string and a character and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a string and a character.

Output consists of a String(the final output).

Refer sample output for formatting specifications.

Sample Input:

Rabbit

-

Sample Output:

t-i-b-b-a-R

ANSWER:

```
using System;




class Program

{

    public static void Main( string[] args )

   {

          string str=Console.ReadLine();
      char ch = Convert.ToChar(Console.ReadLine());
          Console.WriteLine(UserProgramCode.reshape(str,ch));
      Console.Read();




          }

 }
```

```
----------------------

USER PROGRAM

----------------------

using System;


class UserProgramCode

{


   public static string reshape(string str, char ch)

   {


      int l = str.Length;

      string sree = "";

      char[] temp = str.ToCharArray();

      for (int i = l - 1; i >= 0; i--)

      {

         sree = string.Concat(sree, temp[i]);

         if (i != 0)

         {


            sree = string.Concat(sree, ch);

         }

      }
```

```
        return (sree);
    }
}
```

27. SumOfSquaresOfDigits

QUESTION:

Sum of Squares of Digits

Write a program to find out the sum of squares of digits in a given number.

Example:

input = 321

output = (3*3+2*2+1*1) = 14

Include a class UserProgramCode with a static method getSumOfSquaresOfDigits that accepts an integer and returns an integer.

Create a class Program which would get the input and call the static method getSumOfSquaresOfDigits present in the UserProgramCode.

Input and Output Format: Input consists of an Integer. Output consists of an integer.

Sample Input 1: 123 Sample Output 1: 14 Sample Input 2: 102 Sample Output 2: 5

ANSWER:

```
using System;

using System.Collections.Generic;

using System.Linq;
```

```
using System.Text;



class Program

{

   static void Main(string[] args)

   {

      int n = Convert.ToInt32(Console.ReadLine());

      Console.WriteLine(UserProgramCode.getSumOfSquaresOfDigits(n));
      Console.ReadKey();
}
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
class Program

{

   static void Main(string[] args)

   {

      int n = Convert.ToInt32(Console.ReadLine());

      Console.WriteLine(UserProgramCode.getSumOfSquaresOfDigits(n));
      Console.ReadKey();

}

}



class UserProgramCode
{
   public static int getSumOfSquaresOfDigits(int n)
   {
```

```
                    // fill your code here
          int sum = 0, squareDigit, digit, num;
      num = n;
      while (num != 0)
      {
         digit = num % 10;
         squareDigit = digit * digit;
         sum = sum + squareDigit;
         num = num / 10;
      }
      return sum;
           }
}
```

28. Electricity Bill

QUESTION:

Calculate Bill

Write a program to calculate the bill given the previous reading , current reading and per unit charge as inputs.

Example:

input1 =ABC2012345

input2 = ABC2012660

input3 = 4

Bill = (12660 - 12345) * 4

output = 1260

Include a class UserProgramCode with static method calculateBill() that accepts 2 strings corresponding to the previous reading and current reading and an integer that corresponds to the per unit charge. This method returns an integer that corresponds to the bill amount to be paid. Create a class Program which would get the inputs and call the static method calculateBill() present in the UserProgramCode.

Input and Output Format:

Reading Format - XXXXXAAAAA where XXXXX is consumer number and AAAAA is meter reading.

Input1 is a String - previous reading of the consumer

Input2 is a String - current reading of the consumer Input3 is an integer - per unit charge to the consumer output is an integer - Calculated BILL value.

Metric BILL Formula: Bill=(current reading-previous reading)*per unit charge

Sample Input 1: ABC2012345 ABC2012660 4 Sample Output 1: 1260

ANSWER:

qn..29

```
class Program
  {
     static void Main(string[] args)
```

```
    {
        string s1 = Console.ReadLine();
        string s2 = Console.ReadLine();
        int n = int.Parse(Console.ReadLine());
        int sl = UserProgramCode.calculateBill(s1,s2,n);
        Console.WriteLine(sl);
        Console.ReadLine();
    }
}
class UserProgramCode
{
    public static int calculateBill(string s1,string s2,int n)
    {
        int sum = 0;
        string ss1 = s1.Substring(5);
        string ss2 = s2.Substring(5);
        int a = int.Parse(ss1);
        int b = int.Parse(ss2);
        sum = (b - a) * n;
        return sum;
    }
}
```

29. Check Batch Code

QUESTION:

Check Batch Code

Write a program which will check if the given input string follows the below format and print the output according to the conditions given below. 1. The format of the string should be 'AAABBCCXXX' where a. AAA represents the location of the batch CHN -- Chennai CBE -- Coimbatore KOC - Kochi PUN - Pune BGL - Bangalore HYD - Hyderabad KOL - Kolkata Business rules: THe characters 'AAA' should not be other than the above specified values(Only Capitals). If it is other than these characters, print -1. b. BB and XXX in the format represents numerals between 0-9. BB Represents the year and XXX represents the batch code.If other than these are present print -2. c.CC in the format should be only 'DN', if not print -3. All the characters in the input string are in upper case. Please make sure you dont do a spell mistake in the output string. Example 1: Input : CHN13DN014 The output string should be in the following format. DotNet batch 014 has joined in 2013 year and is at Chennai Location

Create a class named UserProgramCode that has the following static method

public static string checkBatch(string input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of a string.

Refer business rules and sample output for formatting specifications.

Sample Input 1 :

CHN13DN014

Sample Output 1 :

DotNet batch 014 has joined in 2013 year and is at Chennai Location Sample Input 2 :

PUN13DN004

Sample Output 2 : DotNet batch 004 has joined in 2013 year and is at Pune Location Sample Input 3 :

BGL14DN014

Sample Output 3 :

DotNet batch 014 has joined in 2014 year and is at Bangalore Location

ANSWER:

5.STRING BATCH CODE:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;



namespace ConsoleApplication90

{

   class Program

   {

      static void Main(string[] args)

      {

         string s = Console.ReadLine();

         var x = Class1.batch(s);

         Console.WriteLine(x);

         Console.ReadLine();

      }

   }

}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;

namespace ConsoleApplication90
{
   class Class1
   {
      public static string batch(string s)
      {
         string op = "";
         Regex re = new Regex(@"^([A-Z]{3}[0-9]{2}[D][N][0-9]{3})$");
         if (re.IsMatch(s))
         {
            string loc = s.Substring(0, 3);
            string output = "";
            switch (loc)
            {
               case "CHN":
                  output = "Chennai";
                  break;
```

```
case "CBE":
   output = "Coimbatore";
   break;
case "KOC":
   output = "Kochi";
   break;
case "PUN":
   output = "PUNE";
   break;
case "BGL":
   output = "BANGALORE";
   break;
case "HYD":
   output = "HYDERABAD";
   break;
case "KOL":
   output = "KOLKATTA";
   break;
 default:
   output = "-1";
   break;
}
```

```
            op = "DotNet batch" +" "+ s.Substring((s.Length - 3))+" " + "has joined in" +" " +"20" +
s.Substring(3, 2) + "year and is at"+" "+ output +" "+ "Location";


        }
        else
        {
           op = "-4";
        }
        return op;
      }
   }
}
```

30. Relative Order

QUESTION:

Relative Order

Given two input integer arrays input1 and input2, write a program to sort input1 in such a way that the relative order among the elements will be same as those are in input2. For the elements not present in input2,append them at last in sorted order. Business Rules : 1. If any of the given inputs contains any negative number, then print -1. 2. If any of the elements in input 2 array is not available in input 1 array, then print -2. 3. If there are less than 3 elements or more than 15 elements in the input1 array, print -3.

Example

Input Array 1 = {2,1,2,5,7,1,9,3,6,8,8} Input Array 2 = {2,1,8,3| Output Array = {2,2,1,1,8,8,3,5,6,7,9}

Create a class named UserProgramCode that has the following static method

public static int[] relativeOrder(int[] input1,int[] input2) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array 1.

The next 'n' lines of input consist of elements in the input array 1.

The next line of the input consists of an integer, m that corresponds to the number of elements in the input array 2.

The next 'm' lines of input consist of elements in the input array 1.

Refer business rules and sample output for formatting specifications. Sample Input 1 : 11 2 1 2 5 7 1 9 3 6 8 8 4 2 1 8 3 Sample Output 1 : 2 2 1 1 8 8 3 5 6 7 9

Sample Input 2 : 8 2 1 5 7 9 3 6 8 4 2 1 8 3

Sample Output 2 : 2 1 8 3 5 6 7 9 Sample Input 3 : 11 2 1 2 -5 7 1 9 3 6 8 8 4 2 1 8 3

Sample Output 3 :

-1

ANSWER:

program.cs

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace Relative_Order

{

   class Program

   {
```

```
static void Main(string[] args)
{
    int n1 = int.Parse(Console.ReadLine());
    int[] arr1 = new int[n1];
    for (int i = 0; i < n1; i++)
    {

        arr1[i] = int.Parse(Console.ReadLine());

    }
    int n2 = int.Parse(Console.ReadLine());
    int[] arr2 = new int[n2];

    for (int i = 0; i < n2; i++)
    {

        arr2[i] = int.Parse(Console.ReadLine());

    }
    int[] op = UserProgramCode.Relative_order(arr1, arr2);
    {
        foreach (var item in op)
        {
```

```
                Console.WriteLine(item);
            }
        }
    }
}
```

userprogram.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Relative_Order
{
    class UserProgramCode
    {
        public static int[] Relative_order(int[] arr1, int[] arr2)
        {
            List<int> list=new List<int>();
            List<int> list1 = new List<int>();
                for (int i = 0; i < arr2.Length; i++)
```

```
    {
        for (int j = 0; j < arr1.Length; j++)
        {
            if (arr1[j] == arr2[i])
            {
                list.Add(arr1[j]);
            }

        }

    }
for (int i = 0; i < arr1.Length; i++)
                        {
                         if(!list.Contains(arr1[i]))
  {
  list1.Add(arr1[i]);
  }
                        }

    list1.Sort();
    List<int> finalList = new List<int>(list);
    for (int i = 0; i < list1.Count; i++)
    {
```

```
            finalList.Add(list1[i]);
        }
        return finalList.ToArray();
    }
  }
}
```

31.Relative Order

Given two input integer arrays input1 and input2, write a program to sort input1 in such a way that the relative order among the elements will be same as those are in input2. For the elements not present in input2,append them at last in sorted order. Business Rules : 1. If any of the given inputs contains any negative number, then print -1. 2. If any of the elements in input 2 array is not available in input 1 array, then print -2. 3. If there are less than 3 elements or more than 15 elements in the input1 array, print -3.

Example

Input Array 1 = {2,1,2,5,7,1,9,3,6,8,8} Input Array 2 = {2,1,8,3| Output Array = {2,2,1,1,8,8,3,5,6,7,9}

Create a class named UserProgramCode that has the following static method

public static int[] relativeOrder(int[] input1,int[] input2) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array 1.

The next 'n' lines of input consist of elements in the input array 1.

The next line of the input consists of an integer, m that corresponds to the number of elements in the input array 2.

The next 'm' lines of input consist of elements in the input array 1.

Refer business rules and sample output for formatting specifications. Sample Input 1 : 11 2 1 2 5 7 1 9 3 6 8 8 4 2 1 8 3 Sample Output 1 : 2 2 1 1 8 8 3 5 6 7 9

Sample Input 2 : 8 2 1 5 7 9 3 6 8 4 2 1 8 3

Sample Output 2 : 2 1 8 3 5 6 7 9 Sample Input 3 : 11 2 1 2 -5 7 1 9 3 6 8 8 4 2 1 8 3

Sample Output 3 :

-1

program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Relative_Order
{
   class Program
   {
      static void Main(string[] args)
      {
         int n1 = int.Parse(Console.ReadLine());
         int[] arr1 = new int[n1];
         for (int i = 0; i < n1; i++)
         {

            arr1[i] = int.Parse(Console.ReadLine());

         }
         int n2 = int.Parse(Console.ReadLine());
         int[] arr2 = new int[n2];

         for (int i = 0; i < n2; i++)
         {

            arr2[i] = int.Parse(Console.ReadLine());
```

```
        }
        int[] op = UserProgramCode.Relative_order(arr1, arr2);
        {
           foreach (var item in op)
           {
              Console.WriteLine(item);
           }
        }
     }
  }
}
```

userprogram.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Relative_Order
{
   class UserProgramCode
   {
      public static int[] Relative_order(int[] arr1, int[] arr2)
      {
         List<int> list=new List<int>();
         List<int> list1 = new List<int>();
            for (int i = 0; i < arr2.Length; i++)
            {
               for (int j = 0; j < arr1.Length; j++)
               {
                  if (arr1[j] == arr2[i])
```

```
                {
                   list.Add(arr1[j]);
                }

              }

          }
        for (int i = 0; i < arr1.Length; i++)
                                {
                                 if(!list.Contains(arr1[i]))
         {
         list1.Add(arr1[i]);
         }
                                }

           list1.Sort();
           List<int> finalList = new List<int>(list);
           for (int i = 0; i < list1.Count; i++)
           {
              finalList.Add(list1[i]);
           }
           return finalList.ToArray();
      }
   }
}
```

32. Repeat Characters

Write a program to repeat the string multiple times provided with the below limitations. a. Consider the index position of the input word starts with 1. b. If the Input1 string length is odd, then the even index position characters should be removed from the input string and the remaining characters should be repeated based on Input2 value. c. If the Input1 string length is even then the odd index position characters should be removed from the input string and the remaining characters should be repeated based on Input2 value. Business Rules : 1. If the Input2 value is negative, then print “Invalid Input”. 2. If

the Input2 value is greater than 10, then print “Input value is too long”. 3. If the length of Input1 is less than 2, then print “Input value is insufficient”.

Create a class named UserProgramCode that has the following static method

public static string repeatRemoveString(string input1, int input2) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of a string and an integer.

Output is a string.

Refer business rules and sample output for formatting specifications. Sample Input 1 : Price 4

Sample Output 1 :

PiePiePiePie Sample Input 2 :

A 8

Sample Output 2 :

Input value is insufficient

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace Repeated_Characcters

{

   class Program

   {

      static void Main(string[] args)

      {

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

```
        int n = int.Parse(Console.ReadLine());
        int len = str.Length;
        string op = UserProgramCode.repeatRemoveString(str, n);
        if (op == "-1")
           Console.WriteLine("nvalid Input");
        else if (op == "-2")
           Console.WriteLine("Input Value is too long");
        else if (op == "-3")
           Console.WriteLine("Input value is insufficient");
        else

        for (int i = 0; i < n; i++)
        {
           Console.Write(op);
        }

        Console.ReadLine();
      }
   }
}
using System;
using System.Collections.Generic;
using System.Linq;
```

```
using System.Text;


namespace Repeated_Characcters
{
   class UserProgramCode
   {
      public static string repeatRemoveString(string input1, int input2)
      {
         int len = input1.Length;
         string s = "";
         if (input2 < 0)
         {
            return "-1";
         }
         else if (input2 > 10)
         {
            return "-2";


         }
         else if (len < 2)
         {
            return "-3";
         }
```

```
else if(len % 2 != 0)
{
   for (int i = 0; i < input1.Length; i++)
   {
      if (i % 2 == 0)
      {
         s = s + input1[i];
      }
   }
}
else
{
   for (int i = 0; i < input1.Length; i++)
   {
      if (i % 2 != 0)
      {
         s = s + input1[i];
      }
   }
}
return s;
```

```
      }
   }
}
```

33. Get the longest string

Write a program to get the longest string from the list which starts with the given character.Assume that input comparison is done irrespective of case. ie case insensitive.

Include a class UserProgramCode with a static method getLongestString which accepts a String list and a character. The return type is a string.

Create a Class Program which would be used to accept the size of the string list, the list elements and the search character and calls the static method present in UserProgramCode.

In getLongestString

1. If there is no element found list, then return the string "No elements found "

2. Only alphabets should be given in the list. Otherwise return the string, "String contains non alphabetic characters. "

3.I f the two or more strings start with the given character ,the longest string should be returned. Assume that the longest string will be unique. Input Output format

First line points to the size of the string list as n.

The next n lines points to elements of the string list.

The last input points to the character.

Output consists of a string. SAMPLE INPUT 1:

4 Yellow Red Black Blue b SAMPLE OUTPUT 1: Black SAMPLE INPUT 2: 3 Black White 45 W SAMPLE OUTPUT 2: String contains non alphabetic characters.

```
using System;
using System.Collections.Generic;
using System.Linq;
```

```
using System.Text;


namespace Get_Logest_String
{
   class UserProgramCode
   {
      public static string LOngeststing(string[] arr, char s)
      {
         List<string> list = new List<string>();
         for (int i = 0; i < arr.Length; i++)
         {
            for (int j = 0; j < arr[i].Length; j++)
            {
               if (!char.IsLetter(arr[i][j]))
               {
                  return "Only alphabets should be";
               }
            }
         }


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

```
    if (arr[i].StartsWith(s.ToString()))
    {
        list.Add(arr[i]);
    }



}


if (list.Count == 0)
    return "No elements Found";
else
{
    int max = 0;
    foreach (string str in list)
    {
        if (str.Length > max)
            max = str.Length;
    }

    foreach (string str in list)
    {
        if (str.Length == max)
        {
```

```
                return str;
            }
        }

    }
    return "";
  }
 }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Get_Logest_String
{
  class Program
  {
    static void Main(string[] args)
    {
      int n = int.Parse(Console.ReadLine());
      string[] arr = new string[n];
```

```
        for (int i = 0; i < n; i++)
        {
            arr[i] = Console.ReadLine();
        }
        Char s = Convert.ToChar(Console.ReadLine());
        string op = UserProgramCode.LOngeststing(arr, s);
        Console.WriteLine(op);
        Console.ReadLine();


    }
  }
}
```

34. Special Characters

Write a program that accepts a string input and removes all the alphabetic characters from input and stores only the special characters and digits. Note: Special characters are #, $,%,& Business Rules : 1. if the given input string contains no numbers or special characters,then print -1. Create a class named UserProgramCode that has the following static method

public static string getSpecialChar(string input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of a string.

Output is either '-1' or the processed string.

Sample Input 1:

cogniz$#45Ant Sample Output 1:

$#45 Sample Input 2: Treasure

Sample Output 2: -1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;
namespace trial
{
   class Program
   {
      static void Main(string[] args)
      {
         String str = Console.ReadLine();
         String res = UserProgramCode.getSpecialChar(str);
         Console.WriteLine(res);
      }
   }
}

//UserProgramCode.cs
using System;
using System.Collections.Generic;
```

```
using System.Linq;
using System.Text;


namespace trial
{
   class UserProgramCode
   {
      public static string getSpecialChar(string input1)
      {
         int num = 0;
         StringBuilder sb=new StringBuilder();
         int sp = 0;
         int len = input1.Length;
         for (int i = 0; i < len; i++)
         {
            char c = input1[i];
            if (char.IsDigit(c))
            {
               num++;
               sb.Append(c);
            }
            if (c == '#' || c == '$' || c == '%' || c == '&')
            {
```

```
            sp++;
            sb.Append(c);
          }
          //else if (!char.IsLetter(c))
          //{
          //   sb.Append(c);
          //}
        }
        if (num == 0 || sp == 0)
        {
          return "-1";
        }
        else
          return sb.ToString();

      }
    }
}
```

35. Arithmetic Operation

Write a program to perform the user specified arithmethic operation. The program will consist of 3 parameters. First one for specifying the type of operation (+,-,*,/) and the other two are the operands upon which the operation has to be performed. Print the final output.

Business Rules:

1. The first parameter should have the values as 1,2,3 or 4. If it has any other value other than this, return -1.

2. The Second and the third parameter should be only positive numbers, else return -2.

3. If the first parameter is

1 ---------- Add the second and third parameter. (second+third)

2 ---------- Subtract the second and third parameter.(second-third)

3 ---------- Multiply second and third parameter. .(second*third)

4 ---------- Divide second by third parameter.(second/third)

Include a class UserProgramCode with a static method arithmeticOperation which accepts three integers. The return type (Integer) should return the result of the operation performed. Return -1 or -2 according to the Business rules.

Create a Class Program which would be used to accept three integers, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of three integers, where the first integer corresponds to the type of operator, the second and third integer corresponds to the operands.

Output consists of an Integer or, a String “Invalid operator” if the -1 is returned, “Invalid operands” if -2 is returned.

Assume all outputs are Integers.

Refer sample output for formatting specifications.

Sample Input 1:

1

2

3

Sample Output 1:

5

Sample Input 2:

5

2

3

Sample Output 2:

Invalid operator

Sample Input 3:

1

-2

3

Sample Output 3:

Invalid operands

qn..22

```
class Program
  {
     static void Main(string[] args)
     {
        int a = int.Parse(Console.ReadLine());
        int b = int.Parse(Console.ReadLine());
        int c = int.Parse(Console.ReadLine());
        int s1 = UserProgramCode.arithmeticOperation(a,b,c);
```

```
        if (s1 == -1)
            Console.WriteLine("Invalid Operator");
        else if (s1 == -2)
            Console.WriteLine("Invalid Operands");
        else
        Console.WriteLine(s1);
        Console.ReadLine();
    }
}
class UserProgramCode
{
    public static int arithmeticOperation(int a,int b,int c)
    {
        if (a > 0 && a < 5)
        {
            if (b < 0 || c < 0)
                return -2;
            else
                if (a == 1)
                    return b + c;
                else if (a == 2)
                    return b - c;
                else if (a == 3)
```

```
                return b * c;
            else
                return b / c;
        }
        else
            return -1;
    }
}
```

36. Get Middle Characters

Write a program to read a string of even length and to fetch two middle most characters. Print the output.

Example:

Input = this

Output1 = hi

Include a class UserProgramCode with a static method getMiddleChars which accepts a String. The return type (String) should return the output String.

Create a Class Program which would be used to accept a String, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a String of even length.

Output consists of a String, the middle two letters

Refer sample output for formatting specifications.

Sample Input 1:

This

Sample Output 1:

hi

qn..23

```
class Program
  {
     static void Main(string[] args)
     {
        string str=Console.ReadLine();


        string s1 = UserProgramCode.getMiddleChars(str);
        Console.WriteLine(s1);


        Console.ReadLine();
     }
  }
  class UserProgramCode
  {
     public static string getMiddleChars(string s)
     {
        int i = 0;
        StringBuilder sb = new StringBuilder();
```

```
        if (s.Length % 2 == 0)
        {
           i = s.Length / 2;
           sb.Append(s[i - 1]);
           sb.Append(s[i]);
        }
        return sb.ToString();
     }
  }
```

37. Sum Of Digits

Write a program to read a String and to get the sum of all the digits present in the given string. Print the sum. If there is no digit in the given string print “No digit present”.

Example1:

Input = good23bad4

Output = 2 + 3 + 4 = 9

Include a class UserProgramCode with a static method sumOfDigits which accepts a String. The return type (Integer) should return the sum, or return -1 if no digits are present.

Create a Class Program which would be used to accept a String and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a string.

Output consists of an Integer or a String “No digit present “ ..

Refer sample output for formatting specifications.

Sample Input 1:

good23bad4

Sample Output 1:

9

Sample Input 2:

good@#bad$

Sample Output 2:

No digit present

20)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Fwd_Prgs
{
   public class UserProgramCode
   {
      public static string sumOfDigits(string str)
      {
         char[] s = str.ToCharArray();
         int sum=0;
         for (int i = 0; i < s.Length; i++)
```

```
        if ((s[i] > 47) && (s[i] < 58))
        {
            sum = sum + (int)s[i]-48;

        }
        string res = sum.ToString();
        if (sum == 0)
            return "-1";
        else
            return res;
    }
}

class Program
{
    static void Main(string[] args)
    {
        string s = Console.ReadLine();
        string res = UserProgramCode.sumOfDigits(s);
        if(res=="-1")
            Console.WriteLine("No digit present");
        else
```

```
            Console.WriteLine(res);
        }
    }
}
```

38. Unique Counter

Write a program that reads a string and finds the number of unique characters in the string (ie the number of characters in the string that appear only once in the string). If the given string does not contain any unique characters print “No unique characters”.

Example -

Input: “HelloWorld”

Output: 5

Input: “coco”

Output: “No unique characters”

Include a class UserProgramCode with a static method uniqueCounter which accepts a String. The return type (Integer) should return the number of unique characters or -1.

Create a Class Program which would be used to accept a String, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a String.

Output consists of an Integer(number of unique characters) or a String (“No unique characters” if no unique characters are present).

Refer sample output for formatting specifications.

Sample Input 1:

HelloWorld

Sample Output 1:

5

Sample Input 2:

coco

Sample Output 2:

No unique characters

UNIQue COUNTER

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;

namespace ConsoleApplication20
{
   class Program
   {
      static void Main(string[] args)
      {
         string str1;
         int x;
         str1 = Console.ReadLine();
         x = UserProgramCode.uniqueCounter(str1);
```

```
        if (x == -1)
            Console.WriteLine("No unique characters");
        else
            Console.WriteLine(x);
        Console.Read();
      }
   }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication20

{
   class UserProgramCode
   {
   public static int uniqueCounter(string str)
  {
            int count = 0;
        int count1 = 0;
```

```
char[] ch=str.ToCharArray();
for (int i = 0; i < ch.Length; i++)
{
   count1 = 0;
   char c = ch[i];
   for (int j =0; j < str.Length; j++)
   {
      char cd = ch[j];
      if (cd == c)
      {
         count1++;
      }
   }
   if (count1 == 1)
   {
      count++;
   }
}

if (count == 0)
   return -1;
else
```

```
        return(count);


    }
 }
}
```

39. Test Vowels Order

Write a program that takes a string as input and checks whether the given string contains exactly five vowels and the vowels should be in alphabetical order. Assume there is no repetition of any vowel in the given string. If the above condition is satisfied, print 1. If the above condition is not satisfied, print 2. Business rule: If there is a repetition of vowels in the given string, print -1. Create a class named UserProgramCode that has the following static method

public static int testOrderVowels (string input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of a string.

Output is an integer.

Refer business rules and sample output for formatting specifications. Sample Input :

acebisouzz Sample Output :

1

TestVowelsOrder

```
using System;


class Program
{
```

```
  public static void Main( string[] args )
  {
            string input=Console.ReadLine();
     int result = UserProgramCode.testOrderVowels(input);
     Console.WriteLine(result);
     Console.Read();
  }
}
using System;
using System.Linq;
using System.Text;


class UserProgramCode
{
  public static int testOrderVowels(string input)
  {
     int res=0;
     string result="";
     input.ToLower();
     for (int i = 0; i < input.Length; i++)
     {
        if (input[i] == 'a' || input[i] == 'e' || input[i] == 'i' || input[i] == 'o' || input[i] == 'u')
           result += input[i];
```

```
    }
    if (result.Length>5)
    {
       res =- 1;
    }
    else if (result.Equals("aeiou"))
       res = 1;
    else
       res = 2;
    return res;
  }
}
```

40. Symmetric Difference

Given two integer arrays Input1 and Input2, write a program to calculate the Symmetric Difference of the two input arrays. Symmetric difference is the difference of A Union B and A Intersection B ie. [(A U B) - (A ^ B)] where A is the Input1 array and B is the Input2 array. Union operation(U) merges the two arrays and makes sure that common elements appear only once. Intersection(^) operation includes common elements from both the arrays. Finally sort the output and print the array. Business Rules : 1. If any/all of the Input values in the Input1 array is negative, then print -1. 2. If any/all of the input values in the Input2 array is negative, then print -2. 3. If all the integers in Input1 array is common to Input2 array, then print -3. 4. If none of the integers in Input1 array is common to Input2 array, then print -3. Example 1: input: 5 Input1 : 11 5 14 26 3 input : 3 Input2 : 5 3 1 Output: 1 11 14 26 AUB: {11,5,14,26,3, 1} A^B: {5,3} AUB - A^B = {1, 11, 14, 26} Example 2: Input : 3 Input1 : 2 16 -9 Input : 3 Input2 : 53 43 31 Output1: -1

Create a class named UserProgramCode that has the following static method

public static int[] symmetricDifference(int[] input1, int[] input2)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input1 array.

The next 'n' lines of input consist of elements in the input1 array.

The next line of the input consists of an integer, m that corresponds to the number of elements in the input2 array.

The next 'm' lines of input consist of elements in the input2 array.

Refer business rules and sample output for output formatting specifications.

Sample Input : 5 11 5 14 26 3 3 5 3

1 Sample Output : 1 11 14 26

```
using System;

using System.Linq;

using System.Collections.Generic;

using System.Text;

class UserProgramCode

{

   public static int[] symmetricDifference(int[] input1, int[] input2)

   {

      var union = input1.Union(input2);

      var intersect = input1.Intersect(input2);

      var symmetric = union.Except(intersect);

      int[] result = new int[symmetric.Count()];

      // Write intersection to screen.
```

```
        int i = 0;
        foreach (int value in symmetric)
        {
            result[i] = value;
            i++;
        }
        return result;
    }
}
---------
Program
---------
using System;
using System.Linq;
using System.Collections.Generic;
using System.Text;

class Program
{
    public static void Main(string[] args)
    {

        int input1, input2;
```

```
input1 = Convert.ToInt32(Console.ReadLine());



int[] inputArr1 = new int[input1];

//int[] output = new int[10];

for (int i = 0; i < input1; i++)
{

    inputArr1[i] = Convert.ToInt32(Console.ReadLine());

}
input2 = Convert.ToInt32(Console.ReadLine());



int[] inputArr2 = new int[input2];

for (int i = 0; i < input2; i++)
{

    inputArr2[i] = Convert.ToInt32(Console.ReadLine());
```

```
    }

    int[] output = UserProgramCode.symmetricDifference(inputArr1, inputArr2);

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

       Console.WriteLine(output[i]);

    }
    Console.Read();

  }
```

41. Gyrating Numbers

Write a program to find whether every integer in a given input integer array is in Gyrating form. Note: Gyrating numbers are numbers whose digits increase and decrease in a continuous repetitive cycle. Every integer of each element should increase or decrease in a continuous sequence.

Business rule: 1) Print 1 if the given input integer array is in Gyrating sequence.

2) Print -1 if the given input integer array is not in Gyrating sequence. 3) Print -2 if the given input integer array consists of a negative number. Example:1 Input: 4 12 321 235 532 Output: 1 Example:2 Input: 4

75 12 531 45 Output: 1 Create a class named UserProgramCode that has the following static method

public static int gyRating(int[] input1)

Create a class named Program that accepts the inputs and calls the static method present in the

UserProgramCode. Input and Output Format: The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.

The next 'n' lines in the input correspond to the elements in the array.

Output is an integer. Refer business rules. Sample Input :

4 12 321 235 532

Sample Output :

1

```
class UserProgramCode
  {
     public int check(int[] a,int len)
     {
        int r,cmax=0,cmin=0,count=0,i=0,rem=0,j=0;
        StringBuilder sb = new StringBuilder();
        foreach (int val in a)
        {
           i = 0; cmax = 0; cmin = 0;
           int v = val;
           if (val < 0)
              return -2;
           else
           {
              while (v > 0)
              {
```

```
++j;
r = v % 10;
if (i == 0)
{
   rem = r;
   ++i;
}
if (r == rem)
{
   ++cmin;
   ++cmax;
}
else if (r < rem)
{
   ++cmin;
   rem = r;
}
else if (r > rem)
{
   ++cmax;
   rem = r;
}
v = v / 10;
```

```
        }
        if (cmin == 1)
            sb.Append("m");
        if (cmax == 1)
            sb.Append("M");
        if (cmin == 1 || cmax == 1 )
        {
            ++count;
        }
        else
            return -1;
    }

}
if (count == len)
{
    //Console.WriteLine("true");
    int counter = 0;
    char odd = sb[1];
    char even = sb[0];
    if (!odd.Equals(even))
    {
        for (int ii = 0; ii < sb.Length; ii++)
```

```
{
   if (ii % 2 == 0)
   {
      //Console.WriteLine("inside even pos");
      if (sb[ii].Equals(even))
         counter++;
      else
         return -1;

   }
   else
   {
      //Console.WriteLine("inside odd pos");
      if (sb[ii].Equals(odd))
         counter++;
      else
         return -1;
   }


}
if (counter == count)
   return 1;
```

```
            }
            else
                return -1;
          }
          return 0;
       }
    }
```

```
#####################
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using ConsoleApplication23;
using ConsoleApplication23;
//using ConsoleApplication9;

namespace ConsoleApplication23
{
class Program
              {
          static void Main(string[] args)
```

```
        {
            UserProgramCode ob = new UserProgramCode();
            int len;
            len = Convert.ToInt32(Console.ReadLine());
            int[] a = new int[len];
            for (int i = 0; i < len; i++)
            {
                a[i] = Convert.ToInt32(Console.ReadLine());
            }
            int res = ob.check(a,len);
            Console.WriteLine(res);
            Console.ReadLine();
        }
    }
}
```

42. Unique Even Sum

Write a program to remove all duplicate elements from an input array and return the sum of all even numbers. Example : Input: {1,2,7,2,4,8,9,6,8} After removing duplicates : {1,7,4,9,6} Output: 4+6 = 10

Exception Rules:

If there is no even number in the input, return -1.

if input contains negative numbers, then return -2.

Include a class UserProgramCode with a static method addUniqueEven which accepts an integer array. The return type is an integer as given in the above statement.

Create a Class Program which would be used to accept Input array and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of n+1 values. The first value corresponds to size of the array. The next n numbers contains the integer values.

Output consists of a integer as mentioned in the problem statement.

Refer sample output for formatting specifications.

Sample Input 1:

9

1

2

7

2

4

8

9

6

8

Sample Output 1:

10

Sample Input 2:

5

1

3

5

7

9

Sample Output 2:

-1

Sample Input 3:

4

1

-2

6

8

Sample Output 3:

-2

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace Unique_Even_Sum
{
   class UserProgramCode
   {
      public static int uniqueevensum(int[] a)
      {
         int flag = 0;


         int len=a.Length;
         int sum = 0;
         for (int i = 0; i <len ; i++)
         {
            if (a[i] < 0)
```

```
        return -2;


}


for (int i = 0; i < len; i++)
{
   if (a[i] % 2 == 0)
   {
       flag = 1;
      break;
   }
}
if (flag == 0)
   return -1;







for (int i = 0; i < len; i++)
{
   for (int j = i+1; j < len; j++)
   {
      if (a[i] == a[j])
```

```
            {


                a[i] = 0;

                a[j] = 0;

            }

          }


        }

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

        {

          if (a[i] % 2 == 0)

          {

            sum = sum + a[i];

          }

        }

        return sum;

      }

   }

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;
```

```
namespace Unique_Even_Sum
{
   class Program
   {
      static void Main(string[] args)
      {
         int n = int.Parse(Console.ReadLine());
         int[] a =new int[n];
         for (int i = 0; i < n; i++)
         {
            a[i] = int.Parse(Console.ReadLine());
         }
         int op = UserProgramCode.uniqueevensum(a);
         Console.WriteLine(op);
         Console.ReadLine();
      }
   }
}
```

43. Check Palindrome

Write a program to read a string input and to check that given string is a palindrome and contains at least two distinct vowels.

The vowels can be repetitive. Even if the same vowel occurs more than once, it should be considered as one vowel only. If the above condition is satisfied, print “Palindrome”, else print “Not Palindrome”.

Include a class UserProgramCode with a static method checkPalindrome which accepts a String. The return type (Integer) should return 1 if Palindrome, else -1.

Create a Class Program which would be used to accept a String, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a string.

Output consists of a String, “Palindrome” or “Not Palindrome”.

Refer sample output for formatting specifications.

Sample Input 1:

himesh

Sample Output 1:

Not Palindrome

Sample Input 2:

ABEBA

Sample Output 2:

Palindrome

33.CHECK_PALLINDROME

```
using System;


class Program
{
   public static void Main(string[] args)
   {
      string str = Console.ReadLine();
```

```
        int output = UserProgramCode.checkPalindrome(str);
        if (output == 1)
            Console.WriteLine("Palindrome");
        else
            Console.WriteLine("Not Palindrome");
    }
}
using System;
using System.Globalization;
using System.Text;

public class UserProgramCode
{
    public static int checkPalindrome(string str)
    {
        char[] a = str.ToCharArray();
        char[] b = str.ToCharArray();
        Array.Reverse(b);
        string stra = new string(a);
        string strb = new string(b);
        if (stra.Equals(strb))
        {
            char[] d = str.ToCharArray();
```

```
        foreach (char item in stra)
        {
            if (item == 'a' || item == 'A' || item == 'e' || item == 'E' || item == 'i' || item == 'I' || item == 'o'
|| item == 'O' || item == 'u' || item == 'U')
            {

                for (int i = 0; i < stra.Length; i++)
                {
                    if (d[i] == item)
                    {
                    }
                    else
                        if (d[i] == 'a' || d[i] == 'A' || d[i] == 'e' || d[i] == 'E' || d[i] == 'i' || d[i] == 'I' || d[i] == 'o'
|| d[i] == 'O' || d[i] == 'u' || d[i] == 'U')
                        {
                            return 1;
                        }
                }
            }
        }
        return -1;
    }
}
```

44. String Array Sorting

Given a string array, write a function to remove the duplicate values from a String Array, sort the strings in ascending and display the string array.

The values 'AA' and 'aa' are NOT the same elements or duplicates. The case sensitive check should be implemented. While sorting, words starting with upper case letters should be considered first.

Business rules:

1) Print 'Invalid String' when the given input integer array consists of any special character or numbers.

2) All the elements in the array should be of same length. If not, then print 'Invalid String'.

Create a class named UserProgramCode that has the following static method

public static string[] orderStringElements(string[] input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format: The first line of the input consists of an integer, n that corresponds to the number of elements in the input string array.

The next 'n' lines in the input correspond to the elements in the string array.

Output is a string array. Refer sample output and business rules Sample Input 1:

6

AAA

BBB

AAA

AAA

CCC

CCC

Sample Output 1:

AAA

BBB

CCC

Sample Input 2:

7

AAA

BBB

aaa

AAA

Abc

A

b

Sample Output 2:

Invalid String

83.

```
//program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;



namespace trial

{

 class Program

{

 static void Main(string[] args)

{
```

```
 int n = int.Parse(Console.ReadLine());
 String[] ar = new String[n];
 for (int i = 0; i < n; i++)
{
 ar[i] = Console.ReadLine();
}
 String[] ret = UserProgramCode.orderStringElements(ar);
 foreach (String a in ret)
{
 Console.WriteLine(a);
}
}
}
}
```

```
//UserProgramCode.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
namespace trial
```

```
{
 class UserProgramCode
{
 public static string[] orderStringElements(string[] ar)
{
 int len = ar[0].Length;
 String pat = @"^[a-zA-Z]{"+len+"}$";
 // Regex reg = new Regex(@"^[a-zA-Z]+$");
 Regex reg1 = new Regex(pat);
 String[] res = new String[1];
 StringBuilder sb = new StringBuilder();
 int n = ar.Length;
 foreach (String aa in ar)
{
 if (!reg1.IsMatch(aa))
{
 res[0] = "Invalid String";
            return res;
}

}
 for (int i = 0; i < n; i++)
{
```

```
 String a = ar[i];


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

{

 if (a.Equals(ar[j]) && !a.Equals(""))

{

 ar[j] = "-1";

}

}

 if (!a.Equals("-1"))

{

 if (i > 0)

            sb.AppendLine();

 sb.Append(a);

}

}

 String[] array = sb.ToString().Split('\n');


Array.Sort(array, StringComparer.Ordinal);

 return array;


}

}
```

}

45. Count Between Numbers Write a program to find the count of elements in the range [l, h] (both inclusive) in an integer array. l corresponds to the lower value and h corrresponds to the higher value. Include a class UserProgramCode with a static method countBetweenNumbers which accepts an integer array and two other integers (l and h). The return type is int. If there are negative numbers in the array or if the value of l or h is negative, return -1. Create a Class Program which would be used to read the inputs and call the static method present in UserProgramCode. If the method returns -1, then print "Negative value Present"

Input and Output Format:

Input consists of n+3 integers. The first integer corresponds to n, the number of elements in the array. The next 'n' integers correspond to the elements in the array. The last integers correspond to the lower value and higher value. Output consists of an integer or a string.

Refer sample output for formatting specifications.

Sample Input 1:

9 2 13 6 19 25 65 34 1 20 5 20

Sample Output 1:

4

Sample Input 2:

2

3 -5

2 3

Sample Output 2:

Negative value Present

Program 53:

==========--

using System;

```
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace progm53
{
   class Program
   {
      static void Main(string[] args)
      {
         int n = Convert.ToInt32(Console.ReadLine());
         int[] a = new int[n];
         for (int i = 0; i < n; i++)
         {
             a[i]=Convert.ToInt32(Console.ReadLine());
         }

         int l = Convert.ToInt32(Console.ReadLine());
         int h = Convert.ToInt32(Console.ReadLine());
         int output=UserProgramCode.countBetweenNumbers(a, l, h);
         if (output != -1)
         {
            Console.WriteLine(output);
```

```
        }
        else
        {
            Console.WriteLine("Negative value Present");
        }
    }
}

class UserProgramCode
{
    public static int countBetweenNumbers(int[] inp,int x,int y)
    {
        int max1=inp.Length;
        int count = 0;
        for (int i = 0; i < max1; i++)
        {
            if ((inp[i] < 0) || (x < 0) || (y < 0))
            {
                count= -1;
            }
            else
            {
                if ((inp[i] >= x) && (inp[i] <= y))
```

```
            {
               count++;
            }
          }
        }
        return count;

    }
  }
}
```

46. String Manipulation

Write a program which can read two strings and add the reverse of the second string in the middle of the first string.

Print "Special character found" if the string consists of special characters, else print the final String.

Examples :

1)

String1 : Arun

String2: Ram

Output : ArmaRun

2)

String1 : Aruns

String2: Ram

Output : ArumaRns

Hint: If the first string contains odd number of characters

e.g. String1 is having 7 characters, then add the second reverse string after the 4 characters of the first

string.

Include a class UserProgramCode with a static method stringManipulation which accepts two Strings. The return type (String) should return the final String.

Create a Class Program which would be used to accept two Strings, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of two Strings.

Output consists of a String(the final String), or “Special character found” if the string consists of special characters.

Refer sample output for formatting specifications.

Sample Input 1:

Arun

Ram

Sample Output 1:

ArmaRun

Sample Input 2:

Aruns

Ram

Sample Output 2:

ArumaRns

Sample Input 3:

Arun$

Ram

Sample Output 3:

Special character found

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication19
{
   class Program
   {
      static void Main(string[] args)
      {
         string ans,str1,str2;
         str1 = Console.ReadLine();
         str2 = Console.ReadLine();
         ans = UserProgramCode.stringmanipulation(str1,str2);

         Console.WriteLine(ans);
         Console.Read();
      }
   }
}
```

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
namespace ConsoleApplication19
{
   class UserProgramCode
   {
      public static string stringmanipulation(string str1, string str2)
      {
         string res1, res2, res3, rev = "";
         int len, len2, len3;
         len = str1.Length;
         if (len % 2 == 0)
         {
            len2 = len / 2;
            res3 = str1.Substring(len2);
         }
         else
         {
            len2 = (len / 2) + 1;
            res3 = str1.Substring(len2);
         }
```

```
        len3 = str2.Length - 1;
        while (len3 >= 0)
        {
           rev = rev + str2[len3];
           len3--;
        }
        res1 = str1.Substring(0, len2);
        res2 = res1 + rev + res3;
        return res2;




      }
   }
}
```

47. Day of Week

Write a program to find out the day of week for given input date which is in string format (MM-dd-yyyy). The output should be in titlecase.

Include a class UserProgramCode with a static method getDay which accepts a string . The return type is string as given in the above statement.

Create a Class Program which would be used to accept Input and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a string.

Output consists of a string as mentioned in the problem statement.

Refer sample output for formatting specifications.

Sample Input 1:

07-13-2012

Sample Output 1:

Friday

Program.Cs

****

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace dateTime
{
   class Program
   {
      static void Main(string[] args)
      {
         string date1 ;
         date1 = Console.ReadLine();
         UserMainCode umc = new UserMainCode();
         Console.WriteLine(umc.timecheck(date1));
         Console.ReadKey();
```

```
        }
    }
}
```

UserMainCode.Cs

****

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Globalization;

namespace dateTime
{
    class UserMainCode
    {
        public string timecheck(string date1)
        {
            DateTime dt1;
            String result = "";
             DateTime.TryParseExact(date1, "MM-dd-yyyy", null, System.Globalization.DateTimeStyles.None,
out dt1);
            result=Convert.ToString(dt1.DayOfWeek);
```

```
        return result;
      }
   }
}
```

48. Find Leaders

Given an array of integer values as input,write a program that will fetch all the leaders in the array and print them after sorting them in ascending order.An element is a leader if it is greater than all the elements to its right side. Consider that the rightmost element is always a leader. Business Rules : 1. If the given input contains any negative number, then print -1. 2. If there are less than 2 elements or more than 10 elements in the input array,print -2. 3. If any of the elements in the input array are repetitive, then print -3. Create a class named UserProgramCode that has the following static method

public static List<int> findLeadersArray(int[] input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.

The next 'n' lines of input consist of elements in the input array.

Refer sample output for formatting specifications.

Sample Input 1: 6 6 7 4 3 5 2

Sample Output 1: 2 5 7

Sample Input 2:

6 6 7 -4 3 5 2

Sample Output 2: -1

173)Leader number

Find leader

```
class UserProgramCode
{
  public static List<int> findLeadersArray(int[] input)
  {
     List<int> l= new List<int>();
     List<int> l1 = new List<int>();
     for (int i = 0; i < input.Length; i++)
     {
        if (input[i] < 0)
        {
           l.Add(-1);
           return l;
        }
     }
     if (input.Length<2 || input.Length>10)
     {
        l.Add(-2);
        return l;
     }
     for (int i = 0; i < input.Length; i++)
     {
```

```
    for (int j = i+1; j < input.Length; j++)
    {
        if (input[i]==input[j])
        {

            l.Add(-3);
            return l;
        }
    }
}

l.Add(input[input.Length - 1]);

int temp = 0;
for (int i = 0; i < input.Length; i++)
{
    for (int j = i+1; j < input.Length; j++)
    {
        if (input[i] > input[j])
        {
            temp = input[i];
        }
        else
```

```
            {
                temp = 0;
                break;
            }
        }
        if (temp != 0)
        {
            l.Add(temp);
        }
        temp = 0;
    }

    l.Sort();
    return l;

 }
class Program
{
  public static void Main( string[] args )
  {
            int size;
            List<int> opli = new List<int>();
            size = Convert.ToInt32(Console.ReadLine());
```

```
        int[] arr = new int[size];
        for(int i=0;i<size;i++)
  {
                arr[i] = Convert.ToInt32(Console.ReadLine());
        }
        opli = UserProgramCode.findLeadersArray(arr);
        for(int i=0;i<opli.Count;i++)
  {
                Console.WriteLine(opli[i]);
        }
  Console.ReadLine();
 }
}
```

49. SortList Write a code to sort the given array of integers in ascending order. Business Rules: Only positive numbers should be given as inputs to the integer array. Include a class UserProgramCode with static method sortList which accepts interger array The return type is a interger array. If the input consists of negative numbers,return -1. If the input array size is 0, return -2.

Create a class Program which would get the input and call the static method sortList present in the UserProgramCode.

If the sortList method returns -1 print "Invalid Input". If the sortList method returns -2 print "Input is Empty". Input Output Format: Input consists of a n+1. n represents the size of the array followed by the elements in the array. Output consists of a array which is sorted in ascending order. Sample Input 1: 3 45 12 36 Sample Output 1 : Sorted Array : 12 36 45 Sample Input 2: 4 -1 56 89 45 Sample Output 2 : Invalid Input

Program 55:

==========

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace progm55
{
   class UserProgramCode
   {
      public static int[] sortList(int[] a)
      {
         int temp;


         int maxi = a.Length;


         if (maxi == 0)
         {
            a[0] = -1;
         }
         else
         {
```

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



      if (a[i] < 0)
      {
         a[0] = -1;


      }


      else
      {
         if (a[i] < a[j])
         {
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;


         }
```

```
            }
          }
        }



      }


         return a;


    }
 }
class Program
 {
    static void Main(string[] args)
    {


       int n = Convert.ToInt32(Console.ReadLine());
       int[] a=new int[n];
       int[] output=new int[n];
      for (int i = 0; i < n; i++)
       {
          a[i] = Convert.ToInt32(Console.ReadLine());
```

```
        }
        output=UserProgramCode.sortList(a);




          if (output[0] != -1)
          {
             for (int i = 0; i < n; i++)
             {
             Console.WriteLine(output[i]);
             }
          }
          else
          {
             Console.WriteLine("Invalid Input");
          }


    }
```

50. Add Series

Write a program that takes an odd positive integer number as input and evaluates the following series: 1+3-5+7-9…+/-n. Example: input = 9 series = 1+3-5+7-9 output = -3

Create a class named UserProgramCode that has the following static method

public static int addSeries(int input1)

Create a class named Program that accepts an integer as input and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of a single integer that corresponds to n.

Output consists of a single integer that corresponds to the sum of the series.

Sample Input: 9

Sample Output : -3

PROGRAM 59.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace level1_59
{
   class Program
   {
      static void Main(string[] args)
      {
         int n,x;
         n = Convert.ToInt32(Console.ReadLine());
         x = UserProgramCode.addSeries(n);
```

```
            Console.WriteLine(x);
        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace level1_59
{
    class UserProgramCode
    {
        public static int addSeries(int a)
        {
            int t = 0, k = 1;
            for (int i = 0; k <= a; i++)
            {
                if (i == 0)
                {
                    t = t + k;
                }
                else if (i == 1)
```

```
            {
               t = t + k;
            }
            else if (i % 2 != 0)
            {
               t = t + k;
            }
            else
            {
               t = t - k;
            }

            k = k + 2;
         }
         return t;
      }
   }
}
```

51. Word Form

Write a program that accepts an integer input and displays the given number in word form.The word form should include only billions,millions,thousands,hundreds wherever applicable. The starting alphabet of each word should be in capital except the word "and". Business Rules: 1)If the given integer is negative convert that to a positive number and append "Minus" before the word then dispaly the result. Create a class named UserProgramCode that has the following static method

public static string wordForm(int number)

Create a class named Program that accepts the input and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of an integer.

Output is a string.

Sample Input 1:

364576567 Sample Output1:

Three Hundred and Sixty Four Million Five Hundred and Seventy Six Thousand Five Hundred and Sixty Seven Sample Input 2: -1234 Sample Output 2:

Minus One Thousand Two Hundred and Thirty Four

79.

//program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace oops2
{

    class Program
    {
        static void Main(string[] args)
        {
```

```
            int num;
            num = int.Parse(Console.ReadLine());
            String ret = UserProgramCode.wordForm(num);
            Console.WriteLine(ret);
        }
    }

}

//UserProgramCode.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace oops2
{
    class UserProgramCode
    {
        public static string wordForm(int n)
        {
            return _toText(n, true);
```

```
    }

    private static string _toText(long n, bool isFirst = false)

    {

        string result;

        if (isFirst && n == 0)

        {

            result = "Zero";

        }

        else if (n < 0)

        {

            result = "Negative " + _toText(-n);

        }

        else if (n == 0)

        {

            result = "";

        }

        else if (n <= 9)

        {

            result = new[] { "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine" }[n - 1] + "
";

        }

        else if (n <= 19)

        {
```

```
            result = new[] { "Ten", "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen",
"Seventeen", "Eighteen", "Nineteen" }[n - 10] + (isFirst ? null : " ");
        }
        else if (n <= 99)
        {
            result = new[] { "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety" }[n /
10 - 2] + (n % 10 > 0 ? "-" + _toText(n % 10) : null);
        }
        else if (n <= 999)
        {
            result = _toText(n / 100) + "Hundred " + _toText(n % 100);
        }
        else if (n <= 999999)
        {
            result = _toText(n / 1000) + "Thousand " + _toText(n % 1000);
        }
        else if (n <= 999999999)
        {
            result = _toText(n / 1000000) + "Million " + _toText(n % 1000000);
        }
        else
        {
            result = _toText(n / 1000000000) + "Billion " + _toText(n % 1000000000);
        }
```

```
        if (isFirst)
        {
            result = result.Trim();
        }
        return result;
    }
  }
}
```

52.

Repeat Characters

Write a program to repeat the string multiple times provided with the below limitations. a. If Input1 string length is five or less than five, then the first three characters should be repeated based on Input2 value. b. If the Input1 string length is more than five then the last three characters should be repeated based on Input2 value Business Rules : 1. If the length of Input1 is less than 3, then print 'Input value is insufficient' 2. If the Input2 value is negative, then print 'Invalid Input' 3. If the Input2 value is greater than 10, then print 'Input value is too long' Example 1: Input1: Price Input2: 3 Output : PriPriPri Example 2: Input1: Sunday Input2: 4 Output: daydaydayday Example 3: Input1: So Input2: 5 Ouput: Input value is insufficient

Create a class named UserProgramCode that has the following static method

public static string repeatManipulateString(string input1, int input2)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of a string.

The second line of input consists of an integer.

Output is a string. Refer sample output and business rules for output formatting specifications.

Sample Input:

Price

3

Sample Output:

PriPriPri

80.

```
//program.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace oops2
{

   class Program
   {
      static void Main(string[] args)
      {
         String val = Console.ReadLine();
         int num = int.Parse(Console.ReadLine());
         String ret = UserProgramCode.repeatManipulateString(val, num);
         Console.WriteLine(ret);
      }
```

```
    }


}
//UserProgramCode.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace oops2
{
    class UserProgramCode
    {
        public static string repeatManipulateString(string input1, int input2)
        {
            StringBuilder sb = new StringBuilder();
            int len = input1.Length;
            if (len < 3)
                return "Input value is insufficient";
            if (input2 < 0)
                return "Invalid Input";
            if (len <= 5)
            {
```

```
            for (int i = 0; i < input2; i++)
            {
                sb.Append(input1.Substring(0,3));
            }
            return sb.ToString();
        }
        else if (len > 10)
        {
            return "Input value is too long";
        }
        else if (len >5)
        {
            for (int i = 0; i < input2; i++)
            {
                sb.Append(input1.Substring(len-3, 3));
            }
            return sb.ToString();
        }
        return " ";
    }
  }
}
```

53. Shipping Cost

Write a program to compute the Cost of Book Shipping. The Shipping Cost is computed according to the shipping type and the package weight. The shipping rate is given below. Shipping types - Weight Rate (bahts/gram) Regular for first 2000 - 0.25 (basic charge) Regular exceeding 2000 - 0.35 for each Express uses the same rate as Regular + 50 bahts fee Note that the Shipping cost is computed from the possible valid minimum rate. Input1- Weight in grams Input2- Type of delivery ('R' Regular and 'X' Express) Create a class named UserProgramCode that has the following static method

public static float CalcShippingCost(float input1, char input2)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of a float that corresponds to the weight in grams.

The second line of the input consists of a character ('R' or 'X') that corresponds to the type of service.

Output consists of a single float that corresponds to the shipping cost. Output is displayed correct to 2 decimal places.

Sample Input:

4500

R

Sample Output:

1375.00

Program 73:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;
```

```
namespace ConsoleApplication18
{
   class Program
   {
      static void Main(string[] args)
      {
         float f=float.Parse(Console.ReadLine());
         char c=Convert.ToChar(Console.ReadLine());
         float p=userProgramCode.CalcShippingCost(f,c);
         Console.WriteLine(p.ToString("F"));
         Console.ReadLine();
      }
   }
   class userProgramCode
   {
      public static float CalcShippingCost(float i, char c)
      {
         float p;
         if (c == 'X')
            i += 50;
         if (i > 2000)
         {
```

```
            float d = i - 2000;
            p =Convert.ToSingle(2000 * 0.25);
            p +=Convert.ToSingle( (d * 0.35));
        }
        else
            p = Convert.ToSingle(i * 0.25);
        return p;
    }


  }
}
```

54. String Processing II

Given a string input input1 ,form another string with the given input string using the following rules. Form the output string with only the last letter of each word of the given input sentence in capital letters separated by $. Dont store $ after the last letter in the output string. Example 1: Input1:This is a cat Output1:SSA$T Example 2: Input1:This7 is a cat Output1: -1 Business Rules : 1. If the given input string contains any number, print -1. 2. If the input contains any special characters, print -2. 3. If there is only one word in input1, then print -3.

Create a class named UserProgramCode that has the following static method

public static string formWordwithLastLetters(string input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format: Input consists of a string.

Output consists of a string or '-1' or '-2' or '-3'. Sample Input 1:

This is a cat

Sample Output 1:

SSA$T

Sample Input 2:

This7 is a cat

Sample Output 2:

-1

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace String_Process2

{

   class UserProgramCode

   {

      public static string StringProcess2(string str)

      {

         string s = "";

         int flag = 0;

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

         {

            if (Char.IsDigit(str[i]))

            {


               flag = 1;
```

```
        return "-1";
    }
}
for (int i = 0; i < str.Length; i++)
{
    if (flag == 0)
    {
        if (!(Char.IsLetterOrDigit(str[i])))
        {
            if (!char.IsWhiteSpace(str[i]))
            {
                flag = 1;
                return "-2";
            }
        }
    }
}
if (flag == 0)
{
    Char[] ch = str.ToCharArray();
    for (int i = 0; i < ch.Length; i++)

    {
```

```
            if(ch[i]==' ')
            {
               flag = 2;


            }
         }
      }
      if (flag != 2)
         return "-3";


      for (int i = 0; i < str.Length - 1; i++)
      {
         if (str[i + 1] == ' ')
         {
            s = s + str[i].ToString().ToUpper() + '$';
         }
      }
      s = s + str[str.Length - 1].ToString().ToUpper();
      return s;
   }
  }
}
```

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace String_Process2

{

   class Program

   {

      static void Main(string[] args)

      {

         string str = Console.ReadLine();

         string op = UserProgramCode.StringProcess2(str);

         Console.WriteLine(op);

         Console.ReadLine();

      }

   }

}
```

55. Calculate Grade

Given an input as integer array with Student_ID and marks as the array element for multiple students in the format. {Student_ID_1, Mark1, Student_ID_2, Mark2, Student_ID_3, Mark3, etc...}, write a program to calculate the grade of the student who has scored the maximum marks and print the output in the

following format.

Student_ID XXX has passed in YYY

where XXX is the Student_ID and YYY is the grade. 1) If Mark is greater than or equal to 80, then store the grade as "DISTINCTION" 2) If Mark is less than 80 and greater than or equal to 60, then store the grade as "FIRST CLASS" 3) If Mark is less then to 60 and greater than or equal to 45 then store the grade as "SECOND CLASS" 4) If Mark is less than 45 and greater than or equal to 0, then store the grade as "FAIL". Business rules: 1) If the Input contains any negative numbers, then print “Invalid Input”. 2) If the number of elements in Input array is less than or equal to 2, then print “Grading is not possible”. 3) If the number of elements in the Input array is odd, then print “Scores not provided for all Students”. Create a class named UserProgramCode that has the following static method
public static string getGrade(int[] input1) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.
Input and Output Format:
The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.
The next 'n' lines of input consist of elements in the input array.
Refer business rules and sample output for formatting specifications. Sample Input 1 : 10 101 80 102 75 103 50 104 60 100 40
Sample Output 1 : Student_ID 101 has passed in DISTINCTION Sample Input 2 : 4 22 9 -5 6
Sample Output 2 :
Invalid Input

17.Calculate Grade

```
using System;


class Program
{
  public static void Main( string[] args )
```

```
 {
            int size;
            size = Convert.ToInt32(Console.ReadLine());
            int[] arr = new int[size];
            for(int i = 0;i<size;i++){
                  arr[i] = Convert.ToInt32(Console.ReadLine());
            }
    string result = UserProgramCode.getGrade(arr);

    Console.WriteLine(result);
    Console.ReadLine();
 }
}
using System;

class UserProgramCode
{
  public static string getGrade(int[] arr)
  {
    int max = arr[1], i;
    int id=arr[0];
    string grade="";
```

```
for (i = 1; i < arr.Length; i=i+2)
{
   if (arr[i] > 0 && arr[i-1]>0 && arr.Length>2 && arr.Length%2==0)
   {
      if (max < arr[i])
      {
         max = arr[i];
         id = arr[i - 1];
      }
   }
   else if(arr[i]<0||arr[i-1]<0)
   {
      grade = "Invalid Input";
      return grade;
   }
   else if (arr.Length<= 2)
   {
      grade = "Grading is not possible";
      return grade;
   }
   else if (arr.Length % 2 != 0)
   {
      grade = "Scores not provided for all Students";
```

```
            return grade;
        }


    }
    if (max >= 80)
    {
        grade = "Student_ID "+id+" has passsed in DISTINCTION";
    }
    else if (max >= 60 && max < 80)
    {
        grade = "Student_ID "+id+" has passed in FIRST CLASS";
    }
    else if (max >= 45 && max < 60)
    {
        grade = "Student_ID "+id+" has passed SECOND CLASS";
    }
    else if (max >= 0 && max < 45)
    {
        grade = "FAIL";
    }
    return grade;
  }
}
```

56. Remove Vowels from Even Position

Write code to remove vowels from even position in the string. Return final string as output. Assume the first character is at position 1 in the given string.

Include a class UserProgramCode with static method removeEvenVowels that accepts the String .The return type is a string.

Create a class Program which would get the input and call the static method removeEvenVowels present in the UserProgramCode.

Input output format Input consists of a string. Output consists of a string. Sample Input 1: commitment Sample Output 1: cmmitmnt Sample Input 2: rythm Sample Output 2: rythm

Program 52:

==========

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace progm52
{
   class Program
   {
      static void Main(string[] args)
      {
         string inpt;
         inpt = Convert.ToString(Console.ReadLine());
```

```
            string outp=UserProgramCode.removeEvenVowels(inpt);
            Console.WriteLine(outp);


        }
    }
class UserProgramCode
    {
        using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace final_f
{
    class UserProgramCode
    {
        public static string removeEvenVowels(string a)
        {
            int maxi = a.Length;


            for (int i = 1; i < maxi; i++)
            {
```

```
            if ((a[i] == 'a') || (a[i] == 'e') || (a[i] == 'i') || (a[i] == 'o') || (a[i] == 'u'))
            {
               a = a.Remove(i, 1);
               maxi--;



            }
         }
         return a;

      }
   }
}


   }


}
```

57. Shipping Cost

Write a program to compute the Cost of Booking for Shipping. The Shipping Cost is computed according to the shipping type and the package weight. The shipping rate is given below.

Shipping types - Weight Rate (bahts/gram)

Regular for first 2000 - 0.25 (basic charge)

Regular exceeding 2000 - 0.35

For each Express, use the same rate as Regular + 50 bahts fee

Note that the Shipping cost is computed from the possible valid minimum rate.

Input1- Weight in grams

Input2- Type of delivery ('R' Regular and 'X' Express)

Example:

Input1: 4500

Input2: R

Output1: 1375

Include a class UserProgramCode with a static method CalcShippingCost which accepts an integer(weight) and a character (type of delivery). The return type (integer) should return the shipping cost.

Create a Class Program which would be used to accept a integer value and a character, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of an integer and a character.

Output consists of an integer(the shipping cost).

Refer sample output for formatting specifications.

Sample input 1:

4500

R

Sample Output 1:

1375

Sample Input 2:

1800

X

Sample Output 2:

500

using System;

```
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Shipping_Cost
{
   class Program
   {
      static void Main(string[] args)
      {
         int f = int.Parse(Console.ReadLine());
         char c = Convert.ToChar(Console.ReadLine());
         int p = UserProgramCode.CalcShippingCost(f, c);
         Console.WriteLine(p);
         Console.ReadLine();
      }
   }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
```

```
using System.Text;


namespace Shipping_Cost
{
   class UserProgramCode
   {
      public static int CalcShippingCost(int i, char c)
      {
         double p;
         if (c == 'X')
            i += 50;
         if (i > 2000)
         {
            int d = i - 2000;
            p = (2000 * 0.25);
            p =p+ ((d * 0.35));
         }
         else
            p = (i * 0.25);
         int a =(int)Math.Round(p,0);


         return a;
      }
```

```
    }
}
```

58. Train Tariff Calculation

Ram has to book his train tickets for travelling from Chennai to Pune through an online portal. The tariffs vary based on parameters such as the Date of booking(DOB) and the Date of Travel(DOT). Consider the number of days between DOB and DOT as NOD. The normal ticket cost from Chennai to Pune when the ticket booking is done minimum one month before is Rs 1000 in Sleeper class(SL). The normal ticket cost for AC class ratings are as follows : First Class AC (1AC):Rs 2500 Second Class AC(2AC): Rs 2000 Third Class AC(3AC) : Rs 1500 A. If NOD is from 21 days upto 30 days ,then the tariff is 10% more than normal ticket cost B. If NOD is from 11 days upto 20 days ,then the tariff is 20% more than normal ticket cost C. If NOD is from 4 days upto 10 days ,then the tariff is 30% more than normal ticket cost D.If NOD is upto 3 days ,then the tariff is 40% more than normal ticket cost. Write a program to calculate the total cost a person has to pay for their booking given their date of booking , the date of travel and the class of travel as input1, input2 and input3 respectively and print the output in the following format. Your ticket is confirmed and the booking cost is Rs YYYYYY where YYYYYY is the calculated booking cost. Print the booking cost as an integer. The date of booking and the date of travel are given as string in the format yyyy.mm.dd Business rules: 1. If the date of travel is less than 3 days from the date of booking, then the tickets cannot be booked and print “Short Notice and hence Tickets cannot be booked” . 2. If the date of booking or the date of travel are not in the date format, then print “Improper Date format in the input” . 3. If the date of travel is more than 90 days from the date of booking, then the tickets cannot be booked and print “Long Notice period and hence Tickets cannot be booked” . 4. If the class of travel is other than SL,1AC,2AC or 3AC, then print “Improper class of Travel” . Create a class named UserProgramCode that has the following static method

public static int calculateTrainTariff(string input1, string input2, string input3) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of 3 strings --- input1 (Date of Booking), input2 (Date of Travel) and input3(class of travel).

Refer business rules and sample output for formatting specifications. Sample Input 1 : 2014.05.15

2014.05.25 SL

Sample Output 1 :

1200 Sample Input 2 : 201405.15 2014.05.25 SL

Sample Output 2 :

Improper Date format in the input

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace ConsoleApplication1

{

   class Program

   {

      static void Main(string[] args)

      {

         string input1, input2, input3;

         input1 = Console.ReadLine();

         input2 = Console.ReadLine();

         input3 = Console.ReadLine();

         int output=UserProgramCode.calculateTrainTariff(input1, input2, input3);

         //output = Math.Round(output, 0);

         if (output > 5)
```

```
{
   Console.WriteLine(output);
}
else
{
   if (output == 1)
   {
      Console.WriteLine("Short Notice and hence Tickets cannot be booked");
   }
   if (output == 2)
   {
      Console.WriteLine("Improper Date format in the input");
   }
   if (output == 3)
   {
      Console.WriteLine("Long Notice period and hence Tickets cannot be booked");
   }
   if (output == 4)
   {
      Console.WriteLine("Improper class of Travel");
   }

}
```

```
            Console.ReadLine();
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class UserProgramCode
    {
        public static int calculateTrainTariff(string input1, string input2, string input3)
        {
            double total = 0;
            try
            {
```

```
DateTime d1 = Convert.ToDateTime(input1);

DateTime d2 = Convert.ToDateTime(input2);


int year = d1.Year - d2.Year;

int month = d1.Month - d2.Month;

int days = d1.Day - d2.Day;

int totaldays = Convert.ToInt32(((d2 - d1).TotalDays)+1);

if (totaldays < 3)

{

   return 1;

}

if (totaldays > 90)

{

   return 3;

}

if ((input3 != "SL") && (input3 != "1AC") && (input3 != "2AC") && (input3 != "3AC"))

{

   return 4;

}

if (totaldays >= 4 && totaldays <= 10)

{

   if (input3 == "SL")

   {
```

```
        total = 1000 * 1.30;
    }
    if (input3 == "1AC")
    {
        total = 2500 * 1.30;
    }
    if (input3 == "2AC")
    {
        total = 2000 * 1.30;
    }
    if (input3 == "3AC")
    {
        total = 1500 * 1.30;
    }

    return (int)total;
}
if (totaldays >= 11 && totaldays <= 20)
{
    if (input3 == "SL")
    {
        total = 1000 * 1.20;
    }
```

```
    if (input3 == "1AC")
    {
        total = 2500 * 1.20;
    }
    if (input3 == "2AC")
    {
        total = 2000 * 1.20;
    }
    if (input3 == "3AC")
    {
        total = 1500 * 1.20;
    }

    return (int)total;
}
if (totaldays >= 21 && totaldays <= 30)
{
    if (input3 == "SL")
    {
        total = 1000 * 1.10;
    }
    if (input3 == "1AC")
    {
```

```
        total = 2500 * 1.10;
    }
    if (input3 == "2AC")
    {
        total = 2000 * 1.10;
    }
    if (input3 == "3AC")
    {
        total = 1500 * 1.10;
    }

    return (int)total;
}
if (totaldays == 3)
{
    if (input3 == "SL")
    {
        total = 1000 * 1.40;
    }
    if (input3 == "1AC")
    {
        total = 2500 * 1.40;
    }
```

```
            if (input3 == "2AC")
            {
               total = 2000 * 1.40;
            }
            if (input3 == "3AC")
            {
               total = 1500 * 1.40;
            }

            return (int)total;
         }
         return 0;
             }
          catch(Exception ex)
          {
             return 2;
          }

      }
   }
}
```

59. Count Vowels Write code to check total number of vowels in the given string. Example: input = ""avinash"" output = 3 Include a class UserProgramCode with static method countVowles which accepts string and return an interger value. Create a class Program which would get the input and call the static

method countVowles present in the UserProgramCode.

Input and Output Format: Input consists of a string Output consists of a integer which corresponds to the number of vowels in the input string. Sample Input 1: suraj Sample Output 1: 2 Sample Input 2: why Sample Output 2: 0

53.CountVowels

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace CountVowels
{
   class Program
   {
      static void Main(string[] args)
      {
         string str = Console.ReadLine();
         int rslt = UserProgramCode.countVowels(str);

         if (rslt == -1)
            Console.WriteLine("Other character found");
         else
```

```
            Console.WriteLine(rslt);



        Console.ReadLine();
      }
   }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;

namespace CountVowels
{
   class UserProgramCode
   {

      public static int countVowels(string str)
      {
         string str1=str.ToLower();
         int rs=0;
         Regex reg = new Regex("^[a-z]+$");
```

```
        if (reg.IsMatch(str1))
        {
           foreach (char c in str1)
           {
              if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u')
              {
                 rs++;
              }
           }
           // return rs;

        }
        else
        {
           rs = -1;
        }

        return rs;
     }
  }
}
```

60. Image Types

Given a string array input which consists of image file names along with their respective image type extensions in the format ("filename.extensiontype",..so on). The image file name and the extension are

seperated by a dot (.)operator. Write a program to calculate the count of image files having same extension type and store the values in the output string array variable in the below format. output (Key,Value) = (ExtensionType1,count1,ExtensionType2,count2,...so on) . Output should be stored in descending order based on the count of image files having the same extension type. Note: jpeg,jfif,exif,tiff,raw,gif,bmp,png are the various types of image file extensions Business Rules: 1)If all the elements of the input array do not have image type extension, then print -1. 2)If any of the file name doesn't contain extension type or if the extension is not an image type then it will be treated as other type, take the count of all such files and store as the last element in the sorted output array with key element as "others" and value element as the calculated count. 3)If more than one key element have same count, then store the key and their respective value element in the order given in input.

Create a class named UserProgramCode that has the following static method

public static List<string> countImageTypes(string[] input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array .

The next 'n' lines of input correspond to elements in the input array.

Refer business rules and sample output for formatting specifications. Sample Input 1 :

4 Employee.jpeg Purchase.jpeg stock.jpeg book.gif

Sample Output 1 : jpeg 3 gif 1 Sample Input 2 :

7

Sales.doc Employee.jpeg Purchase.jpeg image.png stock.jpeg book.gif pen

Sample Output 2 : jpeg 3 png 1 gif 1 others 2

36)image types

using System;

using System.Collections.Generic;

```
using System.Linq;
using System.Text;


namespace image
{
   class Program
   {
      static void Main(string[] args)
      {
         int a = int.Parse(Console.ReadLine());
         List<string> image = new List<string>();
         for (int i = 0; i < a; i++)
         {
            image.Add(Console.ReadLine());



         }
         List<string> ouyp = userProgramcode.imagescount(image);


         foreach (string s in ouyp)
         {


            Console.WriteLine(s);
```

```
        }
        Console.ReadLine();

    }

  }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace image
{
  class userProgramcode
  {
    public static List<string> imagescount(List<string> input1)
    {
      int k = 0,ctr=0;
      string[] ar=new string[8]{"jpeg","jfif","exif","tiff","raw","gif","bmp","png"};
      List<string> outp = new List<string>();
```

```
string[] st = new string[input1.Count];
int[] sco = new int[input1.Count];
for (int j = 0; j < input1.Count; j++)
{
   string[] arr = input1[j].Split('.');
   if (arr.Length==2 && !st.Contains(arr[1]) && ar.Contains(arr[1]) )
   {

      st[k] = arr[1];
      sco[k] = sco[k] + 1;
      k++;
   }
   else if (arr.Length == 2 && st.Contains(arr[1]) && ar.Contains(arr[1])  )
   {
      for (int p = 0; p < st.Length; p++)
      {
         if (st[p] == arr[1])
         {
            sco[p] = sco[p] + 1;
            break;
         }
      }
   }
```

```
    else
    {
        ctr++;

    }
}

if (ctr != input1.Count)
{
    int[] co = new int[k];
    co = sco.ToArray();
    sco = co.Distinct().ToArray();
    Array.Sort(sco);
    Array.Reverse(sco);

    for (int m = 0; m < sco.Length - 1; m++)
    {
        for (int n = 0; n < co.Length; n++)
        {
            if (sco[m] == co[n])
            {
                outp.Add(st[n]);
                outp.Add(sco[m].ToString());
```

```
            }
          }
        }
      }
      if (ctr != 0 && ctr!=input1.Count)
      {
        outp.Add("Others");
        outp.Add(ctr.ToString());
      }
      else if (ctr == input1.Count)
      {
        outp.Add("-1");
        return outp;
      }

      return outp;
    }
  }

    }
```

61.Find Average

Write a program to read an Integer (the size of the List) and the List of Integers and find the average of the numbers as a float value. Print the average.

Print Error Code “Negative numbers present” when inputs other than positive numbers is given.

Include a class UserProgramCode with a static method findAverage which accepts an Integer list. The return type (Float) should return the average value. If negative numbers are present in the array, then return -1.

Create a Class Program which would be used to accept an Integer and an Integer list, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of n+1 Integers, where the first number corresponds the size of the array, followed by the array elements.

Output consists of a Float, the average value, or a String “Negative numbers present” if a negative number is present in the array.

Refer sample output for formatting specifications.

Sample Input 1:

4

2

3

2

3

Sample Output 1:

2.5

Sample Input 2:

2

1

-2

Sample Output 2:

Negative numbers present

46.find average:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication18
{
   class UserProgramCode
   {
      public static float compute(int[] array, int size)
      {
         float avg,sum = 0;
         int i;
         foreach (int a in array)
         {
            if (a < 0)
               return -1;
         }
       for (i = 0; i < size; i++)
```

```
            {
                sum = sum + array[i];
            }
            avg = sum / size;

            return avg;
        }
       }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication18
{

    class Program
    {
        static void Main(string[] args)
        {
            UserProgramCode u = new UserProgramCode();
            int n;
```

```
        float avg;
        n = int.Parse(Console.ReadLine());
        int[] a = new int[n];
        for (int i = 0; i < n; i++)
        {
           a[i] = int.Parse(Console.ReadLine());
        }
        avg = UserProgramCode.compute(a, n);
        if (avg == -1)
        {
           Console.WriteLine("Negative numbers present");
        }
        else
        Console.WriteLine(String.Format("{0:0.0}",avg));
      }
    }

}
```

62.Count Characters

Write a program to count the number of characters present in the given input String. Include a class UserProgramCode with static method countCharacters which accepts string array. The return type is a integer value. Create a class Program which would get the input and call the static method countCharacters present in theUserProgramCode .

Sample Input 1: qwerty

Sample Output 1: 6

Sample Input 2: 12345

Sample Output 2: 5

Count Characters:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace ConsoleApplication18
{
   class UserProgramCode
   {
      public static int countcharachters(string[] s)
      {
         int  sum = 0,flag = 0 ;


         foreach (string s1 in s)
         {
            char[] ch = s1.ToCharArray();
            foreach (char c in ch)
            {
```

```
            if (char.IsLetter(c))
            {
               flag++;
            }

          }
        }

        foreach (string s1 in s)
        {
           sum = sum + s1.Length;
        }
        if(flag==sum)
           return sum;

        else
           return -1;

    }
    }
}

using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication18
{

    class Program
    {
        static void Main(string[] args)
        {
            UserProgramCode u=new UserProgramCode();
            int n = int.Parse(Console.ReadLine());
            string[] s=new string[n];
            int result;
            for (int i = 0; i < n; i++)
                s[i] = Console.ReadLine();
            result=UserProgramCode.countcharachters(s);
            if(result==-1)
                Console.WriteLine("Invalid Input");
            else
            Console.WriteLine(result);
        }
```

```
    }
}
}
```

63.Length of the longest string

Write code to find the length of the longest string in the given string list.
Include a class UserProgramCode with static method longestWordLength that accepts the String list and the return type should be int Create a class Program which would get the input and call the static method longestWordLength(String[] array) present in the UserProgramCode. The longestWordLength(String[] array) returns the length of the longest string
Input and Output Format:
The first integer corresponds to n, the number of elements in the list. The next 'n' integers correspond to the elements in the String list. SAMPLE INPUT 1
2
Black
Blue

SAMPLE OUTPUT 1
5

Length of the longest word:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Fwd_Prgs
```

```
{
   public class UserProgramCode
   {
      public static string longestWordLength(string[] s)
      {
         int sum = 0;
         for (int i = 0; i < s.Length; i++)
         {
            if (s[i].Length > sum)
            {
               sum = s[i].Length;
            }
         }
         return sum.ToString();
      }



   }

   class Program
   {
      static void Main(string[] args)
      {
         int n = int.Parse(Console.ReadLine());
         string[] str=new string[n];
         for (int i = 0; i < n; i++)
         {
            str[i] = Console.ReadLine();
         }
         string res = UserProgramCode.longestWordLength(str);
         Console.WriteLine(res);
      }
```

```
    }
}
```

64. Three Digits

Write a program to read a string and check if the given string is in the format "CTS-XXX" where XXX is a three digit number.

Include a class UserProgramCode with a static method validatestrings which accepts a string and returns an integer.. The function returns 1 if the string format is correct, else returns -1.

Create a Class Program which would be used to accept a String and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a string.

Output consists of a string (Valid or Invalid).

Refer sample output for formatting specifications.

Sample Input 1:

CTS-215

Sample Output 1:

Valid

Sample Input 2:

CTS-2L5

Sample Output 2:

Invalid

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
namespace level1_60
{
   class Program
   {
      static void Main(string[] args)
      {
         int x;
         string s = Console.ReadLine();
         x = UserProgramCode.validatestrings(s);
         if (x == 1)
         {
            Console.WriteLine("valid");
         }
         else
         {
            Console.WriteLine("Invalid");
         }

      }
   }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;

namespace level1_60
{
   class UserProgramCode
   {
      public static int validatestrings(string str)
```

```
    {
        int output1 = 0;
        Regex reg = new Regex(@"^([C]+[T]+[S]+[-]+([0-9]{3}))$");
        if (reg.IsMatch(str))
        {

            output1 = 1;
        }

        else
        {
            output1 = -1;
        }
        return output1;

    }
  }
}
```

65. All Vowels

Write a Program to check if given word contains exactly five vowels and the vowels are in alphabetical order. Assume there is no repetition of any vowel in the given string and all letters are in lower case.

Include a class UserProgramCode with a static method testOrderVowels which accepts a string and returns an integer. The method returns 1 if the condition stated above is satisfied. Else the method returns -1.

Create a Class Program which would be used to read a String and call the static method present in UserProgramCode. If the method returns 1, print 'valid'. Else print 'invalid'.

Input and Output Format:

Input consists of a string with maximum size of 100 characters.

Output consists of a single string.

Refer sample output for formatting specifications.

Sample Input 1:

acebisouzz

Sample Output 1:

valid

Sample Input 2:

alphabet Sample Output 2:

invalid

ALL VOWELS.

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;


namespace ConsoleApplication20

{

   class Program

   {

      static void Main(string[] args)

      {

         string input1;

         int output1;
```

```
            input1 = Console.ReadLine();
            output1 = UserProgramCode.testOrderVowels(input1);
            if (output1.Equals(1))
            {
               Console.WriteLine("valid");
            }
            else
            {
               Console.WriteLine("invalid");
            }
            Console.Read();
        }
    }
}



using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication20
```

```
{
    class UserProgramCode
    {
        public static int testOrderVowels(string input1)
        {
            int output1 = 0;

            StringBuilder sb = new StringBuilder();

            char[] ch = input1.ToCharArray();
            for (int i = 0; i < ch.Length; i++)
            {
                if (ch[i] == 'a' || ch[i] == 'e' || ch[i] == 'i' || ch[i] == 'o' || ch[i] == 'u')
                {
                    sb.Append(ch[i]);
                }
                if (sb.ToString() == "aeiou")
                {
                    output1 = 1;
                }
                else
                {
                    output1 = -1;
```

```
            }
          }
          return output1;
       }
    }
}
```

66. String Reversal

Write a program to reverse each word in the given string.
Include a class UserProgramCode with a static method “reverseString” that accepts a string argument and returns a string.
If string contains any special characters then return "-1".
Create a class Program which would get a string as input and call the static method reverseString present in the UserProgramCode. If the method returns -1, then print 'Invalid Input'.
Input and Output Format:
Input consists of a string.
Output consists of a string.
Sample Input 1:
hai hello
Sample Output 1: iah olleh Sample Input 2: how !#$ Sample Output 2: Invalid Input

String reversal

```
using System;
using System.Collections.Generic;
using System.Linq;
```

```
using System.Text;


namespace ConsoleApplication1
{
   class Program
   {
      static void Main(string[] args)
      {


         string s = Console.ReadLine();


         string a = UserProgramCode.reverseString(s);
         if (a == "-1")
         Console.WriteLine("Invalid Input");
         else
         Console.WriteLine(a);
         Console.ReadLine();



      }
   }
   class UserProgramCode
   {
```

```
public static string reverseString(string a)
{
   int l=a.Length;

    if (a.Any(ch => !(Char.IsLetterOrDigit(ch) || char.IsWhiteSpace(ch))))
       return "-1";

   StringBuilder sb = new StringBuilder();
     char[] c;
   string[] s;
   s=a.Split(' ');
   for(int i=0;i<s.Length;i++)
   {
     c= s[i].ToCharArray();

     Array.Reverse(c);
     sb.Append(c);
       sb.Append(" ");
   }

      return sb.ToString();
```

```
    }
  }
}
```

67. Finding common Elements in multiples of 3

Write a program to find the common elements from all the three input integer lists which are also multiple of 3. Sort the result in descending order and print it. Include a class UserProgramCode with static method FindCommonElements() which accepts 3 integer List. The return type is List<int> which returns common elements in the List.

Example : Input :

3 ---------->First list size 21 6 8 2---------->Second list size 21 6 4---------->Third list size 4 89 21 6 Output : 21 6

Create a class Program which would get the input and call the static method FindCommonElements() present in the UserProgramCode.

Input and Output Format: Refer Example for input Format. Output is a list which contains the common elements in multiples of 3 or a string as specified below.

In FindCommonElements()

If there is no common elements found in all of three input lists then assign 0 to the first element of the output list and return the list. If any of the input lists have negative element then then assign -1 to the first element of the output list and return the list. If any of the input lists have element value greater than 500 then assign -2 to the first element of the output list and return the list. In Program class If the method returns a list with the first element being 0, then print "No match found". If the method returns a list with the first element being -1, then print "The list contains negative values". If the

method returns a list with the first element being -2, then print "The elements of the list should be less than or equal to 500". Otherwise print the result.

SAMPLE INPUT 1:

3 21 6 8 2 21 9 4 4 89 21 56 SAMPLE OUTPUT 1: 21

SAMPLE INPUT 2:

4 13 -27 44 9 5 24 9 41 56 8 4 27 24 -9 8 SAMPLE OUTPUT 2: The list contains negative values

SAMPLE INPUT 3:

3 33 27 444 5 24 9 41 56 8 4 27 24 78 55 SAMPLE OUTPUT 3: No match found

SAMPLE INPUT 4:

8 3 666 7 4 9 24 21 8 7 16 17 24 9 21 56 8 6 3 6 7 24 9 8 SAMPLE OUTPUT 4: The elements of the list should be less than or equal to 500

67.Find common elements

FIndCommon

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace fFindCommon
{
   class UserProgramCode
   {
      public static List<int> FindCommon(List<int> list1, List<int> list2)
      {
```

```
List<int> final=new List<int>();
int flag = 0;
int flag1 = 0;
for (int i = 0; i < list1.Count; i++)
{
   if (list1[i] < 0)
   {
      final.Add(-1);
      flag = 1;

   }
}
for (int i = 0; i < list2.Count; i++)
{
   if (list2[i] < 0)
   {
      final.Add(-2);
      flag1 = 2;
   }
}
if (flag == 1 && flag1 == 2)
{
   final.Clear();
```

```
        final.Add(-3);
        return final;
    }
    else if (flag == 1)
        return final;
    else if(flag1==2)
        return final;
    else
    for (int i = 0; i < list1.Count; i++)
    {
        for (int j = 0; j < list2.Count; j++)
        {
            if(list1[i]==list2[j])
            {
                final.Add(list1[i]);
            }
        }
    }

    final.Sort();
    return final;

}
```

```
    }
}



using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace fFindCommon
{
    class Program
    {
        static void Main(string[] args)
        {
            int n1 = int.Parse(Console.ReadLine());
            List<int> list = new List<int>();
            for (int i = 0; i < n1; i++)
            {
                list.Add(Convert.ToInt32(Console.ReadLine()));


            }
            int n2 = int.Parse(Console.ReadLine());
```

```
        List<int> list2 = new List<int>();
        for (int i = 0; i < n2; i++)
        {
           list2.Add(Convert.ToInt32(Console.ReadLine()));
        }

        List<int> op = UserProgramCode.FindCommon(list, list2);
        foreach (int item in op)
        {
           Console.WriteLine(item);
        }
        Console.ReadLine();
      }
   }
}
```

68. MaxMin Sum

Write a program that accepts 3 integer inputs and finds the sum of maximum and minimum. Business Rules : 1) If any/ or all of the input value is negative then print -1. 2) If any two or all the values in the Input are same then print -2. Example 1: Input1: 25 Input2: 2 Input3: 95 Output : 97 (Min 2 + Max 95) Example 2: Input1: -15 Input2: 49 Input3: 5 Output : -1

Create a class named UserProgramCode that has the following static method

public static int sumMaxMin(int input1, int input2, int input3)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format: Input consists of 3 integers.

Output is an integer. Refer sample output and business rules Sample Input 1: 25

2

95

Sample Output 1:

97

Sample Input 2: -15

49

5

Sample Output 2:

-1

68.Maxminsum

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;



namespace ConsoleApplication13
{
   class userprogramcode
   {
```

```
public static int sumMaxMin(int ip1, int ip2, int ip3)
{
   int ans,a,b;
   int[] t1 = new int[3];
   t1[0] = ip1;
   t1[1] = ip2;
   t1[2] = ip3;
   for (int i = 0; i < 3; i++)
      if (t1[i] < 0)
         return -1;
   for (int i = 0; i < 2; i++)
   {
      for (int j = i + 1; j < 3; j++)
      {
         if (t1[i] == t1[j])
            return -2;
      }
   }
   a = t1.Max();
   b = t1.Min();
   ans = a + b;
   return ans;
}
```

```
    }
    class Program
    {
        static void Main(string[] args)
        {


            int x,y,z,k;
            x = Convert.ToInt32(Console.ReadLine());
            y = Convert.ToInt32(Console.ReadLine());
            z = Convert.ToInt32(Console.ReadLine());
           // k = Convert.ToInt32(Console.ReadLine());
            k =userprogramcode.sumMaxMin(x,y,z);
            Console.WriteLine(k);









        }
    }
}
```

69. List the Elements

Write a program that accepts integer list and an integer. List all the elements in the list that are greater than the value of given integer. Print the result in descending order.

Example:
input1: [1,4,7,3,9,15,24]
input2: 17

Output1:[24]

Include a class UserProgramCode with static method GetElements() which accepts an integer list and the integer as input and returns an integer list. If there is no element found in input1, then store -1 to the first element of output list. Create a class Program which would get the input and call the static method GetElements() present in the UserProgramCode. If there is no such element in the input list, print "No element found".

Input and Output Format:
Input consists of n+2 integers. The first integer corresponds to n, the number of elements in the array. The next 'n' integers correspond to the elements in the array.
The last input is an integer.

Output is an integer list or the string "No element found".
Sample Input 1: 7

1
4
7
3
9
15
24
17
Sample Output 1: 24 Sample Input 2: 6 5 9 3 4 16 21 9 Sample Output 2: 21 16

69.List the elements

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace levelI04
{
   class Program
   {
      static void Main(string[] args)
      {
         int n,i;
         n = int.Parse(Console.ReadLine());
         int[] arr = new int[n];
         for (i = 0; i < n; i++)
            arr[i] = int.Parse(Console.ReadLine());
         int limit = int.Parse(Console.ReadLine());
         int[] ans = UserProgramCode.GetElements(arr, limit);
         if(ans[0]==-1)
            Console.WriteLine("No element found");
```

```
        else
        {
           foreach (int item in ans)
           {
              Console.WriteLine(item);
           }
        }

    }
}
class UserProgramCode
{
    public static int[] GetElements(int[] arr,int limit)
    {

        int n = arr.Length;



        int[] temp=new int[n];
        int i,j;



       i=0;
```

```
        for(j=0;j<n;j++)
        {
           if(arr[j]>limit)
           {
              temp[i]=arr[j];
              i++;
           }
        }

        if (temp[0] == 0)
        {
           temp[0] = -1;
           return temp;
        }

        else
        {
           Console.WriteLine("");
           Array.Sort(temp);
           Array.Reverse(temp);
           return temp;
    }
}
```

}

}

70. Reimbursement

Hina University offers fees reimbursement scheme to its students according to the percentage they secure in board examinations, as per the below criteria.

Category A: For 80% to 85% (inclusive of border values) secured in exam: Refundable amount is 40% of the fees paid by the student during the start of the academic year and a cash award.

Category B: For 86% to 90% (inclusive of border values) secured in exam: Refundable amount is 50% of the fees paid by the student during the start of the academic year and a cash award.

Category C: For above 90% : Refundable amount is 60% of the fees paid by the student during the start of the academic year and a cash award.

The University also awards the students with a cash prize of Rs. 3000, Rs. 5000 and Rs. 7000 for Categories A, B and C respectively.

However, for the student to be eligible for reimbursement, he should NOT have a backlog (arrear) in any subject.

Write a program that calculates the total amount the student receives from the University which includes refundable fee amount and cash prize, according to:

1. The fees he pays at the start of the academic year, (this is the first input).
2. His/Her percentage of marks in the exams and (this is the second input).
3. His backlog status(Boolean : True if there is arrear and False if there is no arrear, this is the third input).

(Total amount the student receives = Refundable fee amount + Cash Prize)

Validation Rules :

1. Only positive number greater than 25000 should be given for fees amount else return -1.
2. Only numbers in the range 80 to 100 should be for percentage else return -2.

Include a class UserProgramCode with a static method calulateAmountRefundable which accept the fees, mark percentage and the backlog status. The return type (integer) should return the Refundable amount, or -1, or -2, accordingly.

Create a Class Program which would be used to accept two integers, and a boolean value, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of double ,integer and a boolean value, where double corresponds to the fees, integer corresponds to the percentage and the boolean values corresponds to the backlog status.

Output consists of an Integer or one of the 2 strings ("Low fees amount" or "Invalid percentage").

Refer sample output for formatting specifications.

Sample Input 1:

25000

82

false

Sample Output 1:

13000

Sample Input 2:

20000

82

false

Sample Output 2:

Low fees amount

Sample Input 3:

20000

72

false

Sample Output 3:

Invalid percentage

70.Reimburesment.

```
namespace SM1
{
   class UserProgramCode
   {
      public static int calulateAmountRefundable(double fee, int marks, bool arr)
      {
         double total;

         if (marks < 80 || marks > 100)
            return -2;
         else if (fee < 25000)
            return -1;

         else
         {
            if (!arr)
            {
               if (marks >= 80 && marks <= 85)
               {
```

```
        total = (40 * fee) / 100;
        total = total + 3000;

        return (int)total;
    }
    else if (marks >= 86 && marks <= 90)
    {
        total = (50 * fee) / 100;
        total = total + 5000;

        return (int)total;
    }
    else if (marks >= 90 && marks <= 100)
    {
        total = (60 * fee) / 100;
        total = total + 7000;

        return (int)total;
    }
    else
        return 0;
}
else
```

```
        {
            return -3;
        }
      }

   }




}
class Program
{
   static void Main(string[] args)
   {



      double fees = Convert.ToDouble(Console.ReadLine());
   int mar = Convert.ToInt32(Console.ReadLine());
   bool arrer = (bool)Convert.ToBoolean(Console.ReadLine());
   int res = UserProgramCode.calulateAmountRefundable(fees, mar, arrer);
      if(res==-1)
         Console.WriteLine("Low fees amount");
```

```
        else if(res==-2)
            Console.WriteLine("Invalid percentage");
        else
            Console.WriteLine(res);

        Console.ReadLine();
      }
   }
}
```

71. Next Highest Number

Write a progam that accepts an integer input and finds out all the combinations of the numbers possible with all the digits present in the input integer and then from the list of combinations,picks up the next higher number than the given input and prints it. Business Rules : 1. If the given input integer is a negative number, then print -1 . 2. If the input contains more than 3 digits,then print -2. 3. If any of the digits present in input are found repetitive, then print -3 . Create a class named UserProgramCode that has the following static method

public static int nextHighestNumber(int input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:

Input consists of an integer.

Output consists of an integer.

Sample Input 1:

376 Sample Output 1:

637 Sample Input 2: -236

Sample Output 2: -1

NEXTHIGHESTNUMBER

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;

namespace ConsoleApplication20
{
   class Program
   {
      static void Main(string[] args)
      {
         int n,output;
         n = int.Parse(Console.ReadLine());
         output = UserProgramCode.nextHighestNumber(n);
         Console.WriteLine(output);
         Console.Read();
      }
   }
```

```
}


using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication20

{
   class UserProgramCode
   {
      public static int nextHighestNumber(int n)
      {

         int num = n;



         int l = n.ToString().Length;
         int next;

         if (l == 1)
         {
```

```
    return num;
}
else if (num < 0)
{
    return -1;
}
else if (l > 3)
{
    return -2;
}
else
{
    if (l == 2)
    {
        int rem1 = n % 10;
        n = n / 10;
        next = (rem1 * 10) + n;
        return next;
    }


    else
    {
```

```
int rem1 = n % 10;
n = n / 10;
int rem2 = n % 10;
n = n / 10;
if (rem1 > rem2)
{
   next = (n * 100) + (rem1 * 10) + rem2;
   return next;
}
else if (n < rem1 && rem2 != rem1)
{
   next = (rem1 * 100) + (n * 10) + rem2;
   return next;
}
else if (n < rem2 && rem2 != rem1)
{
   next = (rem2 * 100) + (n) + rem1 * 10;
   return next;
}
else
{
   return -3;
}
```

```
            }


         }


       }
     }

}
```

72.Sum Non Prime Numbers

Write a program to calculate the sum of all the non prime positive numbers less than or equal to the given number.

Note: prime is a natural number greater than 1 that has no positive divisors other than 1 and itself

Example:
input = 9
Prime numbers = 2,3,5 and 7
output = 1+4+6+8+9=28

Include a class UserProgramCode with a static method “addNumbers” that accepts an integer arguement and returns an integer.

Create a class Program which would get an integer as input and call the static method addNumbers present in the UserProgramCode.

Input and Output Format:

Input consists of an integer.

Output consists of an integer.

Sample Input:

9

Sample Output:

28

SumNonPrimeNumbers

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace sum_non_prime
{
   class Program
   {
      static void Main(string[] args)
      {
         int num = int.Parse(Console.ReadLine());
```

```
            int op = UserProgramCode.nonprime(num);
            Console.WriteLine(op);
            Console.ReadLine();
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace sum_non_prime
{
    class UserProgramCode
    {
        public static int nonprime(int num)
        {
            int sum = 1;
            for (int i = 2; i <=num; i++)
            {
```

```
            int c = 0;
            for (int j = 1; j <=i; j++)
            {
               if (i % j == 0)
               {
                  c++;
               }
            }
            if (c != 2)
            {
               sum = sum + i;
            }
         }
         return sum;
      }
   }
}
```

73. Berth Type

Ram books 3 train tickets for his father,grandfather and himself.Their seat numbers are given as input1,input2 and input3 respectively. Each person is assigned a seat based on the berth type preference(Lower [L], Middle [M], Upper [U], SideLower [SL] and SideUpper[SU]). The berth type can be identified based on the following steps: 1. Divide the seat number by 8 2. If the remainder is either 1 or 4, then the berth type is Lower If the remainder is either 2 or 5, then the berth type is Middle If the remainder is either 3 or 6, then the berth type is Upper If the remainder is 7, then the berth type is

SideLower If the remainder is 0, then the berth type is SideUpper 3. Based on the grandfather's berth type, print the following: If berth type is Lower(L), print

Lower berth provided as per request If berth type is other than Lower(L),swap it with other seat numbers which is lower and print

Your seat has been swapped from XX to YY as per preference request If berth type is not Lower(L) for all the three seats,print

Your seat will be changed on the date of travel here XX is the initial seat number of Ram's grandfather and YY is the swapped seat number.First try swapping with Ram's seat if found lower,else with Ram's fathers seat.Maximum seat number limit - 1000. Business rules: 1.If any of the seat number is zero or negative , print “Invalid Seat Number”. 2.If any of the seat numbers contain any alphabets or special characters , print “Invalid Input” Create a class named UserProgramCode that has the following static method

public static string checkBerthType(string input1,string input2,string input3)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of 3 strings – input1 (Ram's father's berth number), input2 (Ram's grandfather's berth number) and input3 (Ram's berth number).

Output consists of a string.

Refer business rules and sample output for formatting specifications.

Sample Input 1 : 76 75 78

Sample Output 1 :

Your seat has been swapped from 75 to 76 as per preference request Sample Input 2 :

76 -75 78

Sample Output 2 :

Invalid Seat Number

BERTH TYPE

```
using System;

using System.Collections.Generic;

using System.Linq;
```

```
using System.Text;


namespace BerthType
{
   class Program
   {
      static void Main(string[] args)
      {


         String s1 = Console.ReadLine();
         String s2 = Console.ReadLine();
         String s3 = Console.ReadLine();
         String r = UserProgramCode.Berth_type(s1, s2, s3);
         Console.WriteLine(r);
         Console.ReadKey();
      }
   }
}
```

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BerthType
{
   class UserProgramCode
   {
      public static String Berth_type(String s1, String s2, String s3)
      {
         String b1 = "", b2 = "", b3 = "";
         int m1, m2, m3, temp = 0;
         String res = "";

         try
         {

            m1 = Convert.ToInt32(s1);
            m2 = Convert.ToInt32(s2);
            m3 = Convert.ToInt32(s3);
            if (m1 < 0 || m2 < 0 || m3 < 0)
            {
               throw new Exception();
```

```
    }
}
catch
{
    res = "Invalid Seat No.";
    return res;
}

int r1 = m1 % 8;
int r2 = m2 % 8;
int r3 = m3 % 8;
if ((r1 == 1 || r1 == 4))
{
    b1 = "Lower";

}
if ((r2 == 1 || r2 == 4))
{
    b2 = "Lower";
}
if ((r3 == 1 || r3 == 4))
{
    b3 = "Lower";
```

```
}
if (b2 == "Lower")
{
   res = "Grandfather got Lower seat";
   return res;
}
else if (b2 != "Lower")
{
   if (b1 == "Lower")
   {
      res = "Your seat has been swapped from " + m2 + " to " + m1;
      temp = m1;
      m1 = m2;
      m2 = temp;
      return res;
   }
   else if (b3 == "Lower")
   {
      res = "Your seat has been swapped from " + m2 + " to " + m3;
      temp = m3;
      m3 = m2;
      m2 = temp;
      return res;
```

```
            }
            else
            {
               res = "Sorry your request can not be processed....";
               return res;
            }

         }
         return res;
      }
   }
}
```

74.BMI CALCULATOR

BMI Calculator Write a program to find the BMI of a person given their height(In Metres) and weight(In Kg) as inputs. Example : input1 = 70 input2 = 1.65 Metres BMI := 70/(1.65*1.65) =25.711 output = Overweight Include a class UserProgramCode with static method BMICalc which accepts two float numbers. The return type is String. Create a class Program which would get the input and call the static method BMICalc present in the UserProgramCode. Input and Output Format: Input1 is a float - Weight(In Kg) Input2 is a float - Height (In Metres) Output is a string – Interpreted BMI value. Metric BMI Formula BMI = (Weight in Kilograms / (Height in Meters x Height in Meters)) Business rule: BMI Interpretation is given below Underweight = BMI of <18.5 Normalweight = BMI of 18.5–24.9 Overweight = BMI of 25–29.9 Obesity = BMI of 30 or greater If zero or negative number is given as input then return "InvalidInput" , otherwise return "Underweight","Normalweight", "Overweight","Obesity" as per Business rule. Sample Input 1: 70 1.65 Sample Output 1: Overweight Sample Input 2: 45 1 Sample Output 2: Obesity

.BMI calculator

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace levelI01
{
   class Program
   {
      static void Main(string[] args)
      {
         float input1, input2;
         input1 = float.Parse(Console.ReadLine());
         input2 = float.Parse(Console.ReadLine());
         Console.WriteLine(UserProgramCode.BMICalc(input1, input2));

      }
   }
   class UserProgramCode
   {
      public static string BMICalc(float input1, float input2)
      {

         float  bmi;

         if (input1 <= 0 || input2 <= 0)
         {
            return("InvalidInput");
         }
         else
```

```
        {

            bmi = (input1 / (input2 * input2));
            if (bmi < 18.5)
                return("Underweight");
            else if (bmi >= 18.5 && bmi <= 24.9)
                return("Normalweight");
            else if (bmi >= 25 && bmi <= 29.9)
                return("Overweight");
            else if (bmi >= 30)
                return("Obesity");
        }
        return ("null");
    }

  }
}
```

75. Validate Voter

Write a program to Validate the eligibility of the users for Voting in Election by accepting the input as Date Of Birth and the Date Of Election. Accept the dates as strings. Validate the DOB against the Date of Election, if it is atleast 18 yrs.

Business Rules:

1.Only date format “mm/dd/yyyy” should be given as input, if not return -1.

2. The eligible voting age is from 18 years.(Including 18)

3. If the age is valid for voting , then return 1.

4. If the age is invalid for voting , then return 0.

Include a class UserProgramCode with a static method validateVoter which accepts two Strings. The return type (Integer) should return a value according to the business rules.

Create a Class Program which would be used to accept two Strings, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of two Strings, the first String corresponds to the DOB and the second String corresponds to the Date Of Election.

Output consists of a String, ("Invalid Date format" if -1 is returned, "Eligible" if 1 is returned, "Not Eligible" if 0 is returned.

Refer sample output for formatting specifications.

Sample Input 1:

12/29/1990

09/11/2014

Sample Output 1:

Eligible

Sample Input 2:

12/29/2010

09/11/2014

Sample Output 2:

Not Eligible

Sample Input 3:

32/29/1990

09/11/2014

Sample Output 3:

Invalid Date format

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ValidateVoter
{
   class Program
   {
      static void Main(string[] args)
      {
         string dob = Console.ReadLine();
         string doe = Console.ReadLine();
         int op = UserProgramCode.ValidateVoter(dob, doe);

         if(op ==1)
            Console.WriteLine("Eligible");
         else if(op==0)
            Console.WriteLine("Not Eligible");
         Console.ReadLine();
```

```
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ValidateVoter
{
    class UserProgramCode

    {
        public static int ValidateVoter(string dob, string doe)
        {
            int output1 = 0;
            DateTime dt;
            DateTime dt1;
```

```
        bool dobres = DateTime.TryParseExact(dob, "dd/mm/yyyy", null,
System.Globalization.DateTimeStyles.None, out dt);
        bool doeres = DateTime.TryParseExact(dob, "dd/mm/yyyy", null,
System.Globalization.DateTimeStyles.None, out dt1);



        if (!(dobres == true && doeres==true))
        {
           int age = dt1.Year - dt.Year;
           if (dt > dt1.AddYears(-age))
           {
              age--;
           }



           if (age >= 18)
           {
              output1 = 1;
           }
           else
           {
              output1 = 0;
           }
```

```
        }



        return output1;
      }
   }
}
```

76. Class Division

Write a program to calculate the division/class obtained by the student when the marks obtained by a student in 5 different subjects are given as inputs.

The student gets a division/class as per the following rules:

Percentage above or equal to 60 - “First Class”.

Percentage between 50 and 59 - “Second Class”.

Percentage between 40 and 49 - “Third Class”.

Percentage less than 40 - “Failed”.

Include a class UserProgramCode with a static method calculateResult which accepts five integers. The return type (String) should return the class of the student.

Create a Class Program which would be used to accept 5 integer inputs and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of five integers.

Output consists of a String(class of the student).

Refer sample output for formatting specifications.

Sample Input 1:

41

45

46

40

41

Sample Output 1:

Third Class

Class Division

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace levelI05
{
   class Program
   {
      static void Main(string[] args)
      {
         int sub1, sub2, sub3, sub4, sub5;
         sub1 = int.Parse(Console.ReadLine());
         sub2 = int.Parse(Console.ReadLine());
```

```
        sub3 = int.Parse(Console.ReadLine());
        sub4 = int.Parse(Console.ReadLine());
        sub5 = int.Parse(Console.ReadLine());

        Console.WriteLine(UserProgramCode.calculateResult(sub1, sub2, sub3, sub4, sub5));
    }
}
class UserProgramCode
{
    public static string calculateResult(int sub1,int sub2,int sub3,int sub4,int sub5)
    {
      // int sub1,sub2,sub4,sub3,sub5;
        int sum=sub1+sub2+sub3+sub4+sub5;

        string str="";
        int avg=sum/5;
     if(avg>=60)
         str="First Class";
     else if(avg>=50 && avg<=59)
         str="Second Class";
     else if(avg>=40 && avg<=49)
         str="Third Class";
     else if(avg<40)
```

```
            str="Failed";
          return str;


    }
    }

}
```

77. Display Students Exam Eligibility status

A university has the following rules for a student to qualify for a degree with A as the main subject and B as the subsidiary subject:

Business Rule:

(a) He/She should get 55 percent or more(>=55%) in A and 45 percent or more(>=45%) in B.
(b) If he/she gets less than 55(<55%) percent in A he/she should get 55 percent or more(>=55%) in B. However, he/she should get at least 45 percent(>=45%) in A.
(c) If he/she gets less than 45 percent(<45%) in B and 65 percent or more(>=65%) in A he/she is allowed to reappear in an examination in B to qualify.
(d) In all other cases he/she is declared to have failed.

Write a code to display the student status according to above rules. Consider inputs as marks for both the subject.

Include a class UserProgramCode with static method FindResult() that accepts two integers. The return type should be String.

Create a class Program which would get the input and call the static method present in the UserProgramCode.

Input and Output Format :

Input1- % of Marks in A

Input2- % of Marks in B

Output1- Result("P" for Pass ,"F" for fail ; "R" for Reappear)

If the input is more than 100 return "Invalid Input" from FindResult(int input1,int input2) else return appropriate result. Then display the result in Program class.

Sample Input 1: 42 45 Sample Output 1: F Sample Input 2: 105 05 Sample Output 2: Invalid Input

STUDENT EXAM ELIGIBILITY

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace StudentExamEligibility
{
   class Program
   {
      static void Main(string[] args)
      {
         int a, b;
         string result;
         a = Convert.ToInt32(Console.ReadLine());
```

```
            b = Convert.ToInt32(Console.ReadLine());
            result = UserProgramCode.FindResult(a, b);
            Console.WriteLine(result);
            Console.ReadKey();
        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace StudentExamEligibility
{
    class UserProgramCode
    {
        public static string FindResult(int a, int b)
        {
            string result;

            if (a > 100 || b > 100)
            {
                result = "Invalid Input";
```

```
    }
    else
    {
        if (a >= 55 && b >= 45)
        {
            result = "P";
        }
        else if ((a < 55 && a >= 45) && b>=55)
        {
            result = "P";
        }
        else if (b < 45 && a >= 65)
        {
            result = "R";
        }
        else
        {
            result = "F";
        }

    }
    return result;
}
```

```
    }
}
```

78.Colour Code

Write a program to find whther the given string corresponds to a valid colour code or not.

Write code to validate the given color code based on following rules:

- Must start with # symbol
- Must contain six characters after #
- It may contain alphabets from A-F (only upper case) or digits from 0-9

Example :

input = #FF9922

output = Valid

Include a class UserProgramCode with a static method validateColorCode. This method returns 1 if the input corresponds to a valid color code. Else this method returns -1.

Create a class Program which would get the input and call the static method validateColorCode present in the UserProgramCode.

Input and Output Format:

Input is a string - color code as value Output is a string - Valid or Invalid Sample Input 1: #FF9922 Sample Output 1: Valid Sample Input 2: 1234567 Sample Output 2: Invalid

```
class Program
  {
     static void Main(string[] args)
     {
        string str = Console.ReadLine();
        int i = UserProgramCode.validateColorCode(str);
```

```
        if (i == 1)
            Console.WriteLine("Valid");
        else
            Console.WriteLine("Invalid");
        Console.ReadLine();
    }

}
class UserProgramCode
{
    public static int validateColorCode(string s)
    {
        int flag = 0;
        if(s.StartsWith("#"))
        {
            if (s.Length == 7)
            {
                char[] ch = s.ToCharArray();
                for(int i=1;i<=6;i++)
                {
                    if (char.IsDigit(ch[i]) || "ABCDEF".Contains(ch[i]))
                    {
                        flag = 1;
```

```
                }
                else
                {
                   flag = 0;
                   break;
                }
             }
          }
       }
       if (flag == 0)
          return -1;
       else
          return 1;
    }
 }
```

79. Reverse the adjacent pairs of letters

Write a program to swap the adjacent letters from the given string. If the string has an odd number of letters, the last letter is unchanged.

Include a class UserProgramCode with static method swapPairs that accepts the string and return type should be string.

Create a class Program which would get the input and call the static method swapPairs present in the UserProgramCode.

If the input contains special characters or numbers, display "Invalid Input" in swapPairs() otherwise display the resultant string.

Input Output format: The input and output should be a String. Sample input 1: Newyork Sample output 1: eNywrok Sample input 2: New!@ Sample output 2: Invalid Input

Reverse the adjacent pairs of letters

```
using System;


namespace myprograms
{
   class Program
   {
      static void Main(string[] args)
      {
         string input = Console.ReadLine();
         Console.WriteLine(UserProgramCode.swapPairs(input));
      }
   }
}
```

```
using System;


namespace myprograms
{
```

```
public class UserProgramCode
{
    public static String swapPairs(string input)
    {
        string str = " ";
        char temp;
        char[] ch = input.ToCharArray();
        for (int i = 0; i < input.Length-1; i++)
        {
            if (!char.IsLetter(ch[i]))
            {
                str = "Invalid Input";
                goto l;
            }
        }
        for (int i = 0; i < input.Length - 1; i++)
        {
            temp = ch[i];
            ch[i] = ch[i + 1];
            ch[i + 1] = temp;
            i = i + 1;
        }
            str = new string(ch);
```

```
        l:return str;


      }
   }
}
      }
   }
}
```

80.Difference between two dates in days

Get two date strings as input and write code to find difference between two dates in days.

Include a class UserProgramCode with a static method getDateDifference which accepts two date strings as input.

The return type of the output is an integer which returns the diffenece between two dates in days.

Create a class Program which would get the input and call the static method getDateDifference present in the UserProgramCode.

Input and Output Format:

Input consists of two date strings.

Format of date : yyyy-mm-dd.

Output is an integer.

Refer sample output for formatting specifications.

Sample Input 1:

2012-03-12

2012-03-14

Sample Output 1:

2

Sample Input 2:

2012-04-25

2012-04-28

Sample Output 2:

3

DIFFERENCE B/W TWO DATES

```
class Program
{
   static void Main(string[] args)
   {
      string s,s1;
      s = Console.ReadLine();
      s1 = Console.ReadLine();
      userprogramcode obj = new userprogramcode();
      Console.WriteLine(obj.getDateDifference(s,s1));

   }
}
```

```
public class userprogramcode
{
   public string getDateDifference(string s,string s1)
   {
      string fm="yyyy-MM-dd";
      DateTime dt1,dt;
      DateTime.TryParseExact(s,fm,null,System.Globalization.DateTimeStyles.None,out dt);
      DateTime.TryParseExact(s1, fm, null, System.Globalization.DateTimeStyles.None, out dt1);

      return Convert.ToString((dt1-dt).Days);
   }
}
```

81. Password Encryption

There is a online shopping site which stores their customer userid and password in their database. However, the password is encrypted before storing them into the DB for maintaining the security. Consider the password and key used for encryption are given as input1 and input2 respectively.Write a function to encrypt based on encryption logic as follows 1) Replace the first letter of every word in input1 with input2. 2) If first letter of any word of input1 matches with input2 then replace the first letter of that word of input1 with the next alphabet and add # symbol next to it. Business Rules : 1)If the password given is empty, then assign Invalid Password to the output1 variable. 2)If the password given consists of digits or special characters,then assign Invalid Input to the output1 variable Create a class named UserProgramCode that has the following static method

public static string replaceChar(String input1,String input2)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format: Input consists of 2 strings. The first string corresponds to the password and the second string corresponds to the key used.

Output consists of a string.

Refer sample output and business rule for output formatting specifications.

Sample Input 1 : Red Green A

Sample Output 1 : Aed Areen Sample Input 2 : Red Sed Yellow

R

Sample Output 2 :

S#ed Red Rellow

password encryption online shop:

program.cs:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace ConsoleApplication223

{

   class Program

   {

      static void Main(string[] args)

      {

         string s1 = Console.ReadLine();
```

```
        string s2 = Console.ReadLine();


        string r = Class1.repl(s1, s2);
        if (r == "-1")
        {
           Console.WriteLine("invalid format");


        }
        else if (r == "-2")
        {
           Console.WriteLine("invalid output");
        }


        Console.ReadLine();


      }
   }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
namespace Password
{
   class UserProgramCode
   {
      public static string repl(string s1, string s2)
      {
         string op = "";
         int count = 0;
         StringBuilder sb = new StringBuilder();
         foreach (char t in s1)
         {
            count++;
         }
         if (count == 0)
         {
            op = "-1";
            return op;
         }
         foreach (char p in s1)
         {
            if ((char.IsDigit(p)))
            //|| (!(char.IsLetterOrDigit(p))))
            {
```

```
        op = "-2";
        return op;

    }
}
string[] l = s1.Split(' ');
for (int i = 0; i < l.Length; i++)
{
    if (l[i].Substring(0, 1).ToString() != s2)
    {

        sb.Append(l[i].Replace(l[i].Substring(0, 1), s2));
        sb.Append(" ");
    }
    else if (l[i].Substring(0, 1) == s2)
    {
        char t = char.Parse(s2);
        t = ++t;
        string temp = t.ToString();
        temp = (char)t + "#";
        sb.Append(l[i].Replace(l[i].Substring(0, 1), temp));
        sb.Append(" ");
```

```
            }


        }


        op = sb.ToString();
        return op;
      }
   }
}
```

82. Odd Even Sum

Write a program to compare the sum of digits at even indexes (say evenSum) and sum of digits at odd indexes (say oddSum) in the given number. Example1: Input1: 23050 (evenSum = 2 + 0 + 0 = 2 oddSum = 3 + 5 = 8) Output = -1 Example2: Input1: 23111 (evenSum = 2 + 1 + 1 = 4 oddSum = 3 + 1 = 4) Output = 1 Business Rule: 1. If both the sums are same then print 1 else print -1.

Create a class named UserProgramCode that has the following static method

public static int sumOfOddEvenPositioned(int input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of an integer.

Output is either 1 or -1.

Sample Input :

23050 Sample Output :

-1

ODD EVENSUM

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace OddEvenSum
{
   class Program
   {
      static void Main(string[] args)
      {
         int num = int.Parse(Console.ReadLine());
         int op = UserProgramcode.SumOddEvenIndex(num);
         Console.WriteLine(op);
         Console.ReadLine();
      }
   }
}
```

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace OddEvenSum
{
   class UserProgramcode
   {
      public static int SumOddEvenIndex(int num)

      {
         int rem = 0,sum1=0,sum2=0,k=1;
         while (num > 0)
         {
            if (k % 2 != 0)
            {
               rem = num % 10;
               sum1 = sum1 + rem;
               num = num / 10;
               k++;
            }
            else
            {
```

```
            rem = num % 10;
            sum2 = sum2 + rem;
            num = num / 10;
            k++;

          }
        }
        if (sum1 == sum2)
          return 1;
        else
          return -1;
      }
   }
}
```

83..Permutations

Given a String as input, write a program to find all possible permutations of the given input. If the string contains duplicate characters, remove the duplicate characters. Sort the array alphabetically and print the resultant array. The strings in input and output are case sensitive. Business Rule: 1. If string contains any special characters or any digits as input, then print “Invalid Input” .

Create a class named UserProgramCode that has the following static method

public static List<string> permString(string input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of a string.

Refer business rules and sample output for formatting specifications.

Sample Input 1 :

cat

Sample Output 1 : act atc cat cta tac tca Sample Input 2 :

ffin

Sample Output 2 : fin fni ifn inf nfi nif Sample Input 3 :

ffin%2

Sample Output 3 : Invalid Input

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace permutations
{
   class UserProgramCode
   {
      public static List<string> FindPermutations(string set)
      {
         var output = new List<string>();
         if (set.Length == 1)
         {
```

```
            output.Add(set);
        }
        else
        {
            var chars = set.ToCharArray();
            foreach (var c in chars)
            {
                var tail = chars.Except(new List<char>() { c });
                foreach (var tailPerms in FindPermutations(new string(tail.ToArray())))
                {
                    output.Add(c + tailPerms);
                }
            }
        }
        return output;
    }
  }
}
```

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace permutations
{
   class Program
   {
      static void Main(string[] args)
      {
         string set = Console.ReadLine();
         List<string> output = UserProgramCode.FindPermutations(set);
         foreach (var s in output)
         {
            Console.WriteLine(s);
         }
         Console.ReadKey();
      }
   }
}
```

84.Shortest Length

Write a method to get the length of the shortest word in the given string array.

Include a class UserProgramCode with a static method shortestWordLength that accepts a string array and returns an integer that corresponds to the length of the shortest word.

Create a class Program which would get the input and call the static method shortestWordLength present in the UserProgramCode.

Input and Output Format: First line of the input consists of an integer that corresponds to the number of elements in the string array. The next n lines of the input consists of the elements in the string array. Assume that all the elements in the string array are single words.

Output is an integer which corresponds to the length of the shortest word Sample Input 1: 3 cherry hai apple Sample Output 1: 3 Sample Input 2: 4 cherry apple blueberry grapes Sample Output 2: 5

```
class Program
  {
    static void Main(string[] args)
    {
      int i = 0;
      int n = int.Parse(Console.ReadLine());
      string[] s = new string[50];
      for (i = 0; i < n; i++)
        s[i] = Console.ReadLine();
      s[i] = "\0";
      int sl = UserProgramCode.shortestWordLength(s);
      Console.WriteLine(sl);
      Console.ReadLine();
    }
  }
```

```
class UserProgramCode
{
    public static int shortestWordLength(string[] s)
    {

        int sl =s[0].Length;
        for (int i = 1; s[i]!="\0"; i++)
        {
            if (s[i].Length < sl)
                sl = s[i].Length;

        }
        return sl;
    }
}
```

85..String Encryption

Write code to encrypt the given string using following rules and print the encrypted string:

Rules:

Replace the characters at odd positions by next character in alphabet. Leave the characters at even positions unchanged. If an odd position charater is 'z' replace it by 'a'. Assume the first character in the string is at position 1. Include a class UserProgramCode with static method encrypt that accepts a string and returns the encrypted string.

Create a class Program which would get the input and call the static method encrypt present in the

UserProgramCode.

Input and Output Format :

The input is a String . The output is a String which holds the encrypted text. Sample Input 1: curiosity

Sample Output 1:

dusipsjtz

```
class Program
  {
     static void Main(string[] args)
     {
        string str = Console.ReadLine();
        string s1=UserProgramCode.method1(str);
        Console.WriteLine(s1);
        Console.ReadLine();
     }
  }
  class UserProgramCode
  {
     public static string method1(string s)
     {
        int i = 0, a = 0;
```

```
        StringBuilder sb = new StringBuilder();
        for (i = 0; i < s.Length; i++)
        {
           if (i % 2 == 0)
           {
              if (s[i] == 'z')
                 sb.Append('a');
              else
              {
                 a = Convert.ToInt16(s[i]);


                 a = a + 1;
                 sb.Append(Convert.ToChar(a));
              }
           }
           else
              sb.Append(s[i]);
         }
        return sb.ToString();
     }
}
```

86.. Postal Tariff

Jack who stays at Delhi sends new year greetings by post to his friends within India. He wants to know the total postal charges he needs to pay for sending the greetings to his friends. There are two types of postal delivery. Normal Post(NP) and Speedy Post (SP). The tariff rates for NP are as follows A. Postal Cost from Delhi to Bhopal(BP) is Rs 100 B. Postal Cost from Delhi to Chennai(CH) is Rs 450 C. Postal Cost from Delhi to Orissa(OS) is Rs 200 For Speedy Post additional 30% of normal Post tariff is charged. The locations and the type of post Jack wants to send are given in the input array where each element of the array is in the format XXYY-where XX represents the location code and YY represents the type of postal delivery done. Write a program to calculate the total cost Jack paid to send the greatings to his friends. Print the output in the following format. Jacks spend Rs ZZZZ to send the greetings where ZZZZ is the total charges calculated. Ignore case sensitivty of input strings. Business rules: 1. If any of the location codes are other than BP,CH or OS,then print "Invalid location Code" . 2. If any of the postal delivery code is other than NP or SP, then print "Invalid Postal Delivery". Create a class named UserProgramCode that has the following static method

public static void getPostalTariff(string[] input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the string array.

The next 'n' lines of input consists of strings that correspond to elements in the string array.

Refer business rules and sample output for output format.

Always display the total charges to be paid as an int.

Sample Input 1:

3 BPSP CHNP BPNP

Sample Output 1: Jack spends Rs 680 to send the greetings Sample Input 2: 3 BPSP CHSP PPNP

Sample Output 2:

Invalid location Code

```
using System;
using System.Text.RegularExpressions;
namespace code1
{
  class Program
{
  static void Main(String[] args)
{
  int n;
n = int.Parse(Console.ReadLine());
  String[] input1=new String[n];


for (int i = 0; i < n; i++)
{
  input1[i] = Console.ReadLine();
}
  UserMainCode.getPostalTariff(input1);









      }
}
```

```
}
```

```
using System;


public class UserMainCode
{


    public static void getPostalTariff(string[] input1)
{
  int length = input1.Length;
  double amount = 0;
  for (int i = 0; i < length; i++)
{
```

```
        if (input1[i].Substring(2, 2) == "SP")
{
  if (input1[i].Substring(0, 2) == "BP")
  amount += (100*1.3);
  else if (input1[i].Substring(0, 2) == "CH")
  amount += (450 * 1.3);
  else if (input1[i].Substring(0, 2) == "OS")
  amount += (200 * 1.3);
  else
  { Console.WriteLine("Invalid location Code"); return; }




        }
  else if (input1[i].Substring(2, 2) == "NP")
{
  if (input1[i].Substring(0, 2) == "BP")
  amount += (100);
  else if (input1[i].Substring(0, 2) == "CH")
  amount += (450);
```

```
 else if (input1[i].Substring(0, 2) == "OS")
 amount += (200);
 else
{Console.WriteLine("Invalid location Code");return;}
}
 else
 { Console.WriteLine("Invalid Postal Delivery"); return;}
}
 Console.WriteLine("Jack spends Rs "+amount+" to send the greetings");
}
}
```

87.. String Finder

Write a program to read three strings which are Searchstring, Str1 and Str2 as input and to find out if Str2 comes after Str1 in the Searchstring. If yes print “Yes” else print “No”.

Example:
input1 = geniousRajKumarDev
input2 = Raj
input3 = Dev
output = Yes

Include a class UserProgramCode with a static method stringFinder which accepts 3 strings. The return type (Integer) should return 1 if the Str2 comes after Str1 in the Searchstring, else return 2.
Create a Class Program which would be used to read 3 strings, and call the static method present in UserProgramCode.
Input and Output Format:

Input consists of three strings which are Searchstring, Str1 and Str2.

Output consists of a String, “Yes” or “No”.

Refer sample output for formatting specifications.

Sample Input 1:

geniousRajKumarDev

Raj

Dev

Sample Output 1:

Yes

Sample Input 2:

geniousRajKumarDev

Dev

Raj

Sample Output 2:

No

string finder

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace ConsoleApplication20
```

```
{
    public class UserMainCode
    {
        public static int stringFinder(string str, string str1, string str2)
        {
            int str1_len = str1.Length;
            int str2_len = str2.Length;
            string temp = "";
            int st = 0, count1 = 0, count2 = 0;
            while (temp != str1)
            {
                temp = str.Substring(st, str1_len);
                st++;
            }
            if (temp == str1)
            {
                count1++;
            }
            // Console.WriteLine(temp + "\t" + st);
            string sub = str.Substring((st + str1_len - 1));
            temp = "";
            st = 0;
            while (temp != str2)
```

```
        {
            temp = str.Substring(st, str2_len);
            st++;
        }
        if (temp == str2)
        {
            count2++;
        }
        if (count1 == 1 && count2 == 1)
            return 1;
        else
            return 2;
        // Console.WriteLine(temp+"\t"+st);
      }
   }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
using System.Text.RegularExpressions;


namespace ConsoleApplication20
{
   class Program
   {
      static void Main(String[] args)
      {
         string str, str1, str2,ans;
         str = Console.ReadLine();
         str1 = Console.ReadLine();
         str2 = Console.ReadLine();
         ans=Convert.ToString(UserMainCode.stringFinder(str, str1, str2));
         if (ans.Equals("1"))
            Console.WriteLine("Yes");
         else
            Console.WriteLine("No");
         Console.Read();


      }
   }
}
```

88..Add non Common Elements

Write a program to read two integer arrays and to add all the non common elements from the 2 integer arrays. Print the final output.

Example:

input1: [7,9,1,0]

input2: [10,6,5]

Output1:38

Business Rules:

Only positive numbers should be given to the input Lists.

1. If the input1 List consists of negative numbers, return -1.
2. If the input2 List consists of negative numbers, return -2.
3. If the both the input lists consists of negative numbers, return -3.

Include a class UserProgramCode with a static method sumNonCommonElement which accepts the inputs in the following order (input1, size1, input2, size2). The return type (integer) should return output according to the business rules.

Create a Class Program which would be used to accept two lists, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of n+m+2 integers, where first two integers corresponds to the size of the two array lists, respectively, followed by the corresponding array elements.

Output consists of an Integer(the corresponding output), or a String “Input 1 has negative numbers” if the first array contains negative numbers, “Input 2 has negative numbers” if the second array contains negative numbers, or “Both inputs has negative numbers” if both array has negative numbers.

Refer sample output for formatting specifications.

Sample Input 1:

4

3

6

9

2

1

10

7

5

Sample Output 1:

40

Sample Input 2:

4

3

-6

9

2

1

10

7

5

Sample Output 2:

Input 1 has negative numbers

Sample Input 3:

4

3

6

9

2

1

10

-7

5

Sample Output 3:

Input 2 has negative numbers

Sample Input 3:

4

3

6

9

-2

1

10

-7

5

Sample Output 3:

Both inputs has negative numbers

.Add non Common Elements

```
using System;

        using System.Collections.Generic;

        using System.Linq;

        using System.Text;

        namespace ConsoleApplication9

        {
```

```
public class UserProgramCode
{
   public static int sumNonCommonElement(int[] ar1,int n,int[] ar2,int m)
   {
      int a=0,b=0,c=0;
      int sum=0;
      int [] temp=new int[m+n];
      //List<int> li=new List<int>;

      for (int i = 0; i < n; i++)
         if (ar1[i] < 0)
            a = 1;
      for (int j = 0; j < m; j++)
         if (ar2[j] < 0)

      if (a == 0 && b == 0)
      {
         for (int i = 0; i < n; i++)
            for (int j = 0; j < m; j++)
               if (ar1[i] == ar2[j])
               {
                  ar1[i] = 0;
                  ar2[j] = 0;
```

```
            }


        return ar1.Sum()+ar2.Sum();



    }
    if (a == 1 && b == 0)
        return -1;
    else if (b == 1 && a == 0)
        return -2;
    if (a == 1 && b == 1)
        return -3;


    return 0;



  }
}
      class Program
      {
        static void Main(string[] args)
        {
```

```
  int n = int.Parse(Console.ReadLine());
  int m = int.Parse(Console.ReadLine());
  int[] ar1=new int[n];
  int[] ar2=new int[m];
  for(int i=0; i<n;i++)
     ar1[i] = int.Parse(Console.ReadLine());
  for (int i = 0; i < m; i++)
     ar2[i] = int.Parse(Console.ReadLine());
  int flag = UserProgramCode.sumNonCommonElement(ar1, n, ar2, m);
  if(flag==-1)
     Console.WriteLine("Input 1 has negative numbers");
  else if(flag==-2)
     Console.WriteLine("Input 2 has negative numbers");
  else if(flag==-3)
     Console.WriteLine("Both inputs has negative numbers");
  else
     Console.WriteLine(flag);


       }
    }
}
```

89.Interleaved Words

Given three input strings input1,input2 and input3, write a program to check whether input3 string is an interleaved word of input1 and input2 strings (i.e. input3 = concatenation of input1 and input2 strings,such that input2 string is concatenated at the end of the input1 string eg. if 'game' is input1 and 'center' is input2,input3 should be equivalent to 'gamecenter' and not 'centergame'). If it's a interleaved word, then print the following : input3 is a interleaved word of input1 and input2 together Replace input1, input2 and input3 with the corresponding inputs. Ignore case sensitiveness in input and use lowercase to print the output. Business rule: 1) If input1 or input2 strings contains any number, then print -1. 2) If both input1 and input2 strings are same, then print -2. 3) If input1 or input2 contains any special characters, then print -3. Create a class named UserProgramCode that has the following static method

public static string checkInterleavedword(string input1,string input2,string input3) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format: Input consists of 3 strings – input1, input2 and input 3.

Refer business rules and sample output for output formatting specifications.

Sample Input 1 : foreign land foreignland

Sample Output 1 :

foreignland is a interleaved word of foreign and land together Sample Input 2 : string1 set string1set

Sample Output 2 :

-1

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace Interleaved

{
```

```
    class Program
    {
        static void Main(string[] args)
        {
            string str1 = Console.ReadLine().ToLower();
            string str2 = Console.ReadLine().ToLower();
            string str3 = Console.ReadLine().ToLower();
            string op = UserProgramCode.checkInterleavedword(str1, str2, str3);
            if (op == "1")
            {
                Console.WriteLine("{0} is a interleaved word of {1} and {2} together", str3, str1, str2);
            }
            else
            Console.WriteLine(op);
            Console.ReadLine();



        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace Interleaved
{
   class UserProgramCode
   {
      public static string checkInterleavedword(string str1, string str2, string str3)
      {
         for (int i = 0; i < str1.Length; i++)
         {
            if (char.IsDigit(str1[i]))
            {
               return "-1";
            }
         }
         for (int i = 0; i < str2.Length; i++)
         {
            if (char.IsDigit(str2[i]))
            {
               return "-1";
```

```
    }
}
if (str1.Equals(str2))
{
    return "-2";
}
for (int i = 0; i < str1.Length; i++)
{
    if (!char.IsLetter(str1[i]))
    {
        return "-3";
    }
}



for (int i = 0; i < str2.Length; i++)
{
    if (!char.IsLetter(str2[i]))
    {
        return "-3";
    }
}
```

```
        if (str3 == str1 + str2)
        {
           return "1";
        }
        return "0";
      }
   }
}
```

90.Cattle Graze

In a village there is a ground with full of grass where the cattle-rearing people take their cattle to maze in the ground. Assume that the cattle is tied to a tree. Write a program to calculate the area of grass that the cattle can maze. The rope length would be the input and area rounded of two decimal places would be the output.

Do not use Math.PI for the value of PI. Use 3.14 directly.

Include a class UserProgramCode with a static method calculateArea which accepts an integer. The return type is double. The method returns the area rounded to 2 decimal places.

Create a Class Program which would be used to accept Input and call the static method present in UserProgramCode. Use random function in Math library.

Input and Output Format:

Input consists of the integer value n.

Output consists of a double.

Refer sample output for formatting specifications.

Sample Input 1:

3

Sample Output 1:

28.26

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Reflection;

namespace ConsoleApplication2
{
   class Program
   {
      static void Main(string[] args)
      {
         int n = int.Parse(Console.ReadLine());
         Console.WriteLine(UserProgramCode.calculateArea(n).ToString("#0.00"));

      }
   }

}

class UserProgramCode
   {
```

```
    public static double calculateArea(int n)
    {
       double area = 0;
       area = Math.Round((3.14*n*n),2);
       return area;

    }
  }
```

91.Form New Word

Write a program to read a string and a positive int (say n) as input and to construct a string with first n and last n characters in the given string. Note - the given string length is >= 2n

Example:

Input1 = California

input2 = 3

output = Calnia

Include a class UserProgramCode with a static method formNewWord() that accepts a string and an integer. The method returns a string. Create a class Program which would get the inputs and call the static method formNewWord() present in the UserProgramCode.

Input and Output Format: Input consists of a string and an integer. Output is a String that corresponds to the newly formed word.

Sample Input 1:

California 3

Sample Output 1:

Calnia

.form new word

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Workout41
{
   class UserProgramCode
   {
      public static string formNewWord(string s,int n)
      {
         string s1,s2;
         int l,n1;
         l = s.Length;
         if (l > n * 2)
         {
            n1 = l - n;
            s1 = s.Substring(0, n) + s.Substring(n1, n);
            return s1;
```

```
        }
        else
        {
            return "";
        }
      }
   }

   class Program
   {
      static void Main(string[] args)
      {
         string s;
         int n;
         UserProgramCode u = new UserProgramCode();
         s = Console.ReadLine();
         n = int.Parse(Console.ReadLine());
         s = UserProgramCode.formNewWord(s,n);
         Console.WriteLine(s);

      }
   }
}
```

91. SORT THE LIST

Write a program which reads an Integer(size of the list), a String List and a character, and to get the strings that will not start with the given character irrespective of case. Sort the elements in ascending order based on its length. Print the output list. If the elements are having the same length, then display the elements in alphabetical order.

Include a class UserProgramCode with static method getTheElements which accepts a String list and a character. The return type is List<String>.

Create a Class Program which would be used to get the inputs and call the static method present in UserProgramCode. In getTheElements method, return "Invalid Input" when any of the input strings contain non-alphabets. When the output list is empty, then return "List is Empty". Otherwise return the appropriate result.

In Program class print the result which is returned by getTheElements method in UserProgramCode. Input output format The first line of the input is an integer that corresponds to n, the size of the list. The next n lines of input correspond to the elements in the string list. The next line of the input contains the character. Sample Input 1: 3 read write edit e Sample Output 1: read write Sample Input2 : 2 Elegent event e Sample Output2 : List is Empty Sample Input 3: 2 Eleg$ent e^ent e Sample Output 3 : Invalid Input

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace new_practice
{
   class Program
   {
      static void Main(string[] args)
      {
```

```
        int n = int.Parse(Console.ReadLine());
        List<string> list = new List<string>();
        for (int i = 0; i < n; i++)
        {
           list.Add(Console.ReadLine());
        }

        char c = char.Parse(Console.ReadLine());
        List<string> op = UserProgramCode.Sort_string(list, c);
        if (list[0] == "-1")
        {
           Console.WriteLine("List is empty");
        }
        else if (list[0] == "-2")
        {
           Console.WriteLine("Invalid Input");
        }
        else
        for (int i = 0; i < op.Count; i++)
        {
           Console.WriteLine(op[i]);
        }
        Console.ReadLine();
      }
   }
}


using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
namespace new_practice
{
   class UserProgramCode
   {
      public static List<string> Sort_string(List<string> list, Char c)
      {
         string temp = "";
         for (int i = 0; i < list.Count; i++)
         {
            if (list[i].StartsWith(c.ToString()))
            {
               list.Remove(list[i]);
               i--;
            }
         }

         if (list.Count == 0)
         {
            list.Add("-1");
            return list;
            //Console.WriteLine("List is empty");
            //Environment.Exit(0);
         }



         for (int i = 0; i < list.Count; i++)
         {
            for (int j = 0; j < list[i].Length; j++)
            {
               if (!char.IsLetter(list[i][j]))
```

```
      {
         list.Clear();
         list.Add("-2");
         return list;
         //Console.WriteLine("Invalid Input");
         //Environment.Exit(0);
      }
   }
}

for (int i = 0; i < list.Count - 1; i++)
{
   for (int j = i + 1; j < list.Count; j++)
   {
      if (list[i].Length > list[j].Length)
      {
         temp = list[i];
         list[i] = list[j];
         list[j] = temp;
      }
      else if(list[i].Length==list[j].Length)
      {
         if (list[i][0] > list[j][0])
         {
            temp = list[i];
            list[i] = list[j];
            list[j] = temp;
         }
      }
   }
}
```

```
            //for (int i = 0; i < list.Count - 1; i++)
            //{
            //    for (int j = i + 1; j < list.Count; j++)
            //    {
            //        if (list[i][0] > list[j][0])
            //        {
            //            temp = list[i];
            //            list[i] = list[j];
            //            list[j] = temp;
            //        }
            //    }
            //}
            return list;
        }
    }
}
```

92. IDENTIFY PERFECT NUMBERS.

For a given integer input array ,write a program to identify the perfect numbers in the input array and store remaining elements excluding the perfect numbers in the output .Perfect number is a positive number in which the sum of all its positive divisors excluding that number is equivalent to that number itself. Eg. 6 is a perfect number ,since its divisor are 1, 2 and 3. Sum of its divisors is 1 + 2+ 3 = 6,which is equal to the number itself. Business rule: 1) If any of the elements in input1 array is negative, then print -1. 2) If there are any duplicates found in input1 array, then print -2. 3) If size of the input1 array is 1 or greater than 7, then print -3. Create a class named UserProgramCode that has the following static method

public static int[] perfectNum(int[] input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.

The next 'n' lines of input consist of elements in the input array.

Refer business rules and sample output for formatting specifications. Sample Input 1 :

4 6 2 5 7

Sample Output 1 : 2 5 7 Sample Input 2 : 5 5 8 3 -4 6

Sample Output 2 : -1

identify perfect

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace perfectnumber
{
   class UserProgramCode
   {

      public static int[] perfect(int[] arr,int n)
           {
              int sum = 0;
         List<int> temp = new List<int>();
         List<int> temp1 = new List<int>();
         if (arr.Length < 1 || arr.Length > 7)
```

```
{
   temp1.Add(-3);
      return temp1.ToArray();
}
for (int i = 0; i < n; i++)
{
   for (int j = 1; j < n; j++)
   {
      if (arr[i] == arr[j])
      {
         temp1.Add(-2);
         return temp1.ToArray();
      }
   }
}



   for (int i = 0; i < n; i++)
   {
      sum = 0;
      if (arr[i] < 0)
      {
         temp1.Add(-1);
```

```
                return temp1.ToArray();

            }

            for (int j = 1; j <= (arr[i] / 2); j++)

            {


                if (arr[i] % j == 0)

                {

                    sum = sum + j;

                }



            }

            if (sum != arr[i])

            {


                temp.Add(arr[i]);




            }

        }

        return temp.ToArray();

    }

}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace perfectnumber
{
   class Program
   {
      static void Main(string[] args)
      {
         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[] op = UserProgramCode.perfect(arr, n);
```

```
            foreach (int item in op)
            {
               Console.WriteLine(item);
            }
            Console.ReadLine();
        }
    }
}
```

93.PASSWORD VALIDATION

Password Validation

Given a method with a password in string format as input, write code to validate the password using following rules:

- Must contain at least one digit
- Must contain at least one of the following special characters @, #, $
- Length should be between 6 and 20 characters (both inclusive).

Include a class UserProgramCode with a static method validatePassword which accepts a password string as input.

If the password is as per the given rules return 1 else return -1.If the return value is 1 then print "Valid password" else print as "Invalid password".

Create a Program class which gets a string as an input and call the static method validatePassword present in the UserProgramCode.

Input and Output Format:

Input is a string .

Output consists of a string. Output "Valid password" if the given password is valid or "Invalid password" if the given password is not valid.

Sample Input 1:

%Dhoom%

Sample Output 1: Invalid password

Sample Input 2:

#@6Don

Sample Output 2:

Valid password

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace level1_56
{
   class Program
   {
      static void Main(string[] args)
      {
         string str1;
         int x;
         str1 = Console.ReadLine();
         x=UserProgramCode.validatePassword(str1);
```

```
        if(x==1)
            Console.WriteLine("valid input");
        else
            Console.WriteLine("Invalid input");
      }
   }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace level1_56
{
   class UserProgramCode
   {
      public static int validatePassword(string str)
      {
         bool a, b, c;
         a = str.Contains("@");
         b = str.Contains("#");
         c = str.Contains("$");
         if (a || b ||s c)
```

```
        {
            if ((str.Length >= 6) && (str.Length <= 20))
            {
                return 1;
            }
        }
        return -1;
    }
  }
}
```

94.COUNT EVEN OCCURRENCE

Count Even Occurrence

Given an int array, write a program to calculate the count as follows. If the same element is repeated even number of times,increase the count by one. Print the value of the count. Business Rules : 1. If any of the element in the array is a negative number, then print -1. 2. If there is no elements repeated in even number of times, then print 0. Create a class named UserProgramCode that has the following static method

public static int countEvenOccurence(int[] input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.

The next 'n' lines of input consist of elements in the input array.

Output consists of an integer.

Refer sample output for formatting specifications.

Sample Input 1 : 17 1 2 3 4 9 3 6 7 1 9 100 2 4 1 45 1 9

Sample Output 1 : 4 Sample Input 2 :

13 1 2 3 4 3 6 7 1 9 100 2 4 -17

Sample Output 2 : -1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace CountEvenOccurance
{
   class Program
   {
      static void Main(string[] args)
      {
         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 op = UserProgramCode.CountEvenOccurance(arr);
```

```
            Console.WriteLine(op);
            Console.ReadLine();

        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace CountEvenOccurance
{
    class UserProgramCode
    {
        public static int CountEvenOccurance(int[] arr)
        {
            for (int i = 0; i < arr.Length; i++)
            {
                if (arr[i] < 0)
```

```
        return -1;
}

int c=0;
int final=0;
for(int i=0;i<arr.Length;i++)
{
   c = 1;
   for(int j=1;j<arr.Length;j++)
   {
      if(arr[i]==arr[j])
      {
         c++;
      }
   }
   if (c % 2 == 0)
   {
      final++;
   }
   else
      return 0;
}
return final;
```

```
        }


      }

    }
```

95.ARRAY MEDIAN

Given an integer array as input, write a program to calculate median of the given numbers in the array with the below conditions 1. Sort the Input array in ascending. 2. When the number of elements in the Input array is odd, then median will be the middle number. Median is the (N+1)/2th element. 3. When the number of elements in the Input array is even, then median will be the average of two middle numbers. Round off the median value to the nearest integer. Business rule: 1) If the Input array contains any negative numbers, then print -1. 2) If any of the Input array elements contains "0", then print -2. Example1: Input : 7 1 2 1 4 7 1 2 Output : 2 After sorting the array is {1,1,1,2,2,4,7} Number of element in input array N is 7 (N+1)/2th element= (7+1)/2= 4th element Median value is the 4th number which is 2 Example 2: Input : 6 52 51 81 84 60 88 Output : 71 After sorting the array is {51,52,60, 81, 84, 88} Median = Average of Middle 2 numbers = (60 + 81)/2 = 70.5. round(70.5) = 71

Create a class named UserProgramCode that has the following static method

public static int calculateMedian(int[] input1) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.

The next 'n' lines of input consist of elements in the input array.

Refer business rules and sample output for formatting specifications.

Sample Input :

7 1 2 1 4 7 1 2

Sample Output :

2

```
------Arrray Median----

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace Arraymedian

{

   class Program

   {

      static void Main(string[] args)

      {

         int x, i;

         double m;

         x =Convert.ToInt32( Console.ReadLine());

         int[] input = new int[x];

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

         {

            input[i]=Convert.ToInt32(Console.ReadLine());

         }

         median c = new median();

         m=c.arraymedian(input);

         double d=Math.Round(m,0,MidpointRounding.AwayFromZero);
```

```
            Console.WriteLine(d);
            Console.ReadLine();
        }
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Arraymedian
{
    class median
    {
        public double arraymedian(int[] input)
        {
            int length;
            double p;
            Array.Sort(input);
            input.Reverse();
            length = input.Length;
            foreach (int i in input)
```

```
    {
      if (i < 0)
      {
        return -1;
      }
      if (i == 0)
      {
        return -2;
      }
    }

    if (length % 2 == 1)
    {
      return input[((length + 1) / 2)-1];
    }
    else
    {
     p= (float)(input[(length / 2)-1] + input[(length / 2)])/2.0;
      return p;
    }

  }
}
```

}

96.REVERSING A STRING

Reversing a String

Given a method that accepts a string and a Character as its input parameters, Write code to reverse the string and return it in a format such that each character is separated by the given character. Create a class named UserProgramCode that has the following static method

public static string reshape(string input1, char input2)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of a string and a character.

Output is a string.

Sample Input :

Rabbit

-

Sample Output :

t-i-b-b-a-R

using System;

do this first

class Program

{

```
    public static void Main( string[] args )

   {

     string str=Console.ReadLine();
      char ch = Convert.ToChar(Console.ReadLine());
                Console.WriteLine(UserProgramCode.reshape(str,ch));
      Console.Read();



     }

  }
```

USER PROGRAM

```
using System;

class UserProgramCode
{

   public static string reshape(string str, char ch)
```

```
   {


      int l = str.Length;

      string sree = "";

      char[] temp = str.ToCharArray();

      for (int i = l - 1; i >= 0; i--)

      {

         sree = string.Concat(sree, temp[i]);

         if (i != 0)

         {


            sree = string.Concat(sree, ch);

         }

      }

      return (sree);

   }

}
```

till here

else down look

==========

```
using System;



class Program
{
  public static void Main( string[] args )
  {
      string inputWord=Console.ReadLine();
      int position=Convert.ToInt32(Console.ReadLine());
      char ch=Convert.ToChar(Console.ReadLine());
      string result=UserProgramCode.replaceString(inputWord,position,ch);
      if(result.Equals("-1"))
                Console.WriteLine("Invalid String");
      else if(result.Equals("-2"))
                Console.WriteLine("Number not positive");
      else if(result.Equals("-3"))
                Console.WriteLine("Character not a special character");
      else
                Console.WriteLine(result);
  }
}
```

```
using System;


class UserProgramCode
{
   public static string replaceString(string inputWord, int position, char ch)
   {
      string inputWord1=inputWord.ToLower();
      foreach (Char z in inputWord)
      {
         if (!(Char.IsLetterOrDigit(z) || Char.IsWhiteSpace(z)))
         {
            return "-1";
         }
      }
      if (position <= 0)
      {
         return "-2";
      }
      if ((Char.IsLetterOrDigit(ch)) || Char.IsWhiteSpace(ch))
      {
         return "-3";
      }
      else
```

```
        {
            string[] A = inputWord1.Split(' ');
            string b = string.Copy(A[position - 1]);
            char[] B = b.ToCharArray();
            for (int i = 0; i < b.Length; i++)
            {
               B[i] = ch;
            }
            string c = new string(B);
            A[position - 1] = c;
            string d = string.Join(" ", A);
            return d;
        }
    }
}
```

97.LARGEST SPAN

Write a program to read the size of the integer array and the elements of the array and find the size of largest Span in the given array. Print the output.

Note: Span is the number of elements between two repeated numbers including both numbers. Assume an array with single element has a span of 1.

Business rule:

If there is no number repeated in an array, return 0. If there are two repeated integers in the input

array, consider the first number and return the span.

Example 1:

Input = {1, 2, 1, 1, 3}

Output = 4

Example 2:

Input = {1, 4, 2, 1, 4, 1, 5}

Output = 6

Include a class UserProgramCode with a static method getMaxSpan which accepts the size of the array and the integer array. The return type (integer) should return the span size.

Create a Class Program which would be used to accept the size of the array and the array elements and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of n+1 integers, where the first integer corresponds to the size of the array followed by n integers .

Output consists of an integer(the span size).

Refer sample output for formatting specifications.

Sample Input 1:

5

1

2

1

1

3

Sample Output 1:

4

Sample Input 2:

7

1

4

2

1

4

1

5

Sample Output 2:

6

NO ANSWER FOUND:::

98. ROMAN NUMERAL

Given an integer as input, write a program to convert integer input to roman numerals . Represent it as a string. Basic Steps for Roman number calculation: 1. I is the numeral one. V is the numeral 5. X is the numeral 10. L is the numeral 50. C is the numeral 100. D is the numeral 500. M is the numeral 1000. 2.A smaller number in front of a larger number means subtraction, everything else means addition. For example, IV means 4, VI means 6. You would not put more than one smaller number in front of a larger number to subtract. For example, IIV would not mean 3. You must separate ones, tens, hundreds, and thousands as separate items. This means that 99 is XCIX, 90 + 9, but never should be written as IC. Similarly, 999 cannot be IM and 1999 cannot be MIM. So, II is two, III is three. VII is 7, VIII is 8. IX is 9, XI is 11, etc. Again, XL would be 40, LX would be 60, LXX would be 70, LXXX would be 80 etc. Similarly, XC would be 90, XCIX would be 99, CL would be 150, CLIX would be 159, CXC would be 190, CC would be 200, CCC would be 300, etc. Again, CD would be 400, DC would be 600, etc. And CM would be 900. Business Rule: 1. If the input is less than 0, then print “Invalid Input” . 2. If the given input variable is greater than 4000, then print “Greater Than 4000” . Create a class named UserProgramCode that has the following static method

public static string romanNumerals(int input1).

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of an integer.

Output is a string.

Refer business rules and sample output for formatting specifications.

Sample Input 1 :

2086

Sample Output 1 :

MMLXXXVI

Sample Input 2 :

2091

Sample Output 2 :

MMXCI

Sample Input 3 :

-2091

Sample Output 3 :

Invalid Input

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace numer
{
   class Program
   {
      static void Main(string[] args)
      {
         int num = int.Parse(Console.ReadLine());
```

```
            string output =UserProgramCode.ToRoman(num);
            Console.WriteLine(output);
            Console.ReadLine();


        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace numer
{
    class UserProgramCode
    {
            public static string ToRoman(int num)
        {
            if (num > 3999) throw new ArgumentException("Too big - can't exceed 3999");
            if (num < 1) throw new ArgumentException("Too small - can't be less than 1");
            int thousands, hundreds, tens, units;
            thousands = num / 1000;
            num %= 1000;
```

```
            hundreds = num / 100;
            num %= 100;
            tens = num / 10;
            units = num % 10;
            var sb = new StringBuilder();
            if (thousands > 0) sb.Append(roman1[3 - thousands]);
            if (hundreds > 0) sb.Append(roman2[9 - hundreds]);
            if (tens > 0) sb.Append(roman3[9 - tens]);
            if (units > 0) sb.Append(roman4[9 - units]);
            return sb.ToString();
        }
        static string[] roman1 = { "MMM", "MM", "M" };
        static string[] roman2 = { "CM", "DCCC", "DCC", "DC", "D", "CD", "CCC", "CC", "C" };
        static string[] roman3 = { "XC", "LXXX", "LXX", "LX", "L", "XL", "XXX", "XX", "X" };
        static string[] roman4 = { "IX", "VIII", "VII", "VI", "V", "IV", "III", "II", "I" };

    }
}
```

99.DIFFERENCE BETWEEN DATES IN MONTHS

Find the difference between Dates in months

Given a method with two date strings in yyyy-mm-dd format as input, write code to find the difference between two dates in months.

Include a class UserProgramCode with a static method getMonthDifference which accepts two date strings as input. The method returns an integer which is the difference between two dates in months.

Create a class Program which would get the input and call the static method getMonthDifference present in the UserProgramCode.

Input and Output Format:
Input consists of two date strings.
Format of date : yyyy-mm-dd.

Output is an integer.
Refer sample output for formatting specifications.

Sample Input 1:
2012-03-01
2012-04-16
Sample Output 1:
1

Sample Input 2:
2011-03-01
2012-04-16
Sample Output 2:
13

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
namespace level1_58
{
   class Program
   {
      static void Main(string[] args)
      {
         int k;
         string intime, outtime;
         intime = Console.ReadLine();
         outtime = Console.ReadLine();
         k = UserProgramCode.getMonthDifference(intime, outtime);
         if (k == -1)
         {
            Console.WriteLine("Invalid format");
         }
         else
            Console.WriteLine(k);


      }
   }
}
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace level1_58
{
   class UserProgramCode
   {
      public static int getMonthDifference(string intime, string outtime)
      {

         string s;
         int d1 = 0;
         s = "yyyy-MM-dd";
         DateTime i, o;
         bool k = DateTime.TryParseExact(intime, s, null,
System.Globalization.DateTimeStyles.None, out i);
         if (k == false)
            return -1;
         bool j = DateTime.TryParseExact(outtime, s, null,
System.Globalization.DateTimeStyles.None, out o);
         if (j == false)
            return -1;
         int a1 = i.Month;
```

```
        int a2 = o.Month;
        int d;
        if (a1 > a2)
           d = a1 - a2;
        else
           d = a2 - a1;
        int y1 = i.Year;
        int y2 = o.Year;
        if (y1 > y2)
           d1 = y1 - y2;
        else
           d1 = y2 - y1;
        return ((d1*12)+d);


      }
    }
}
```

100.PRINT CAPITALIZED

Write a code to convert the first letter of each word to capital Case and return the final string Example : Input: ""Now is the time to act!"" Output: ""Now Is The Time To Act!"" Include a Class UserProgramCode with a static method printCapitalized which accepts a string as an input.The return type is String which is a sentence with first letter of each word capitalized. Create a Class Program which would be used to accept String and call the static method present in UserProgramCode. Input and Output Format: Input

consists of string. Output consists of a string which corressponds to first letter of each word to be capitalized and make other letters to be lower case

Sample Input 1:

Features Of JAVA2

Sample Output 1:

Features Of Java2

Sample Input 2:

gOogLe is A SeaRCh enGinEe

Sample Output 2:

Google Is A Search Enginee

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace ConsoleApplication2
{
   class Program
   {
      static void Main(string[] args)
      {
         string input,output;
         input = Console.ReadLine();
         Class1 c = new Class1();
         output=c.capitals(input);
```

```
            Console.WriteLine(output);
            Console.ReadLine();

        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication2
{
    class Class1
    {
        public string capitals(string input)
        {
            string output1;
            char c;
            int i = 0,k=0;
            string[] s = new string[10];
```

```
        char[] ch = new char[10];
        string sout;
        StringBuilder sb = new StringBuilder();
        s = input.Split(' ');
        foreach (string str in s)
        {
            c = str[0];
            ch = str.ToCharArray();
            k = 0;
            foreach (char d in ch)
            {
                ch[k]=char.ToLower(d);
                k = k + 1;
            }
            ch[0] = char.ToUpper(c);
            sout = new string(ch);
            sb.Append(sout);
            sb.Append(' ');
            i = i + 1;
        }
        return sb.ToString();
    }
}
```

}

101.NUMBER AVAILABILITY

Write the program to find whether the given number is available in a list of numbers.

Get three input parameters, one the size of the list, second the list of numbers and the other the given number to be searched. Print the output as - “Non Positive”, “Present”, or “Not Present” respectively as per the given business rules.

Business Rules:

1.List of numbers and the number to be searched, all of them should be positive numbers only, if not return -1.

2. If the given number is present in the list of numbers , then return 1.

3. If the given number is not present in the list of numbers , then return 0.

Include a class UserProgramCode with a static method findExistence which accepts the size of the integer array, an integer array and the number to be searched. The return type (Integer) should return -1, 1 or 0 as per the given business rules.

Create a Class Program which would be used to accept the size of the array, the array elements and an integer, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of an integer which corresponds to the size of the array, an integer array and an integer.

Output consists of a String(“Non Positive”, “Present”, or “Not Present”).

Refer sample output for formatting specifications.

Sample Input 1:

3

1

2

3

1

Sample Output 1:

Present

Sample Input 2:

3

-1

2

3

3

Sample Output 2:

Non Positive

Sample Input 3:

3

1

2

3

4

Sample Output 3:

Not Present

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace ConsoleApplication99
{
   class Program
   {
      static void Main(string[] args)
```

```
    {
        int[] a = new int[30];
        int l = 1;
        for (int i = 0; i <=l+1; i++)
        {
            a[i] = Convert.ToInt16(Console.ReadLine());
            l = a[0];
        }
        int b = UserProgramCode.findExistence(a);
        if(b==1)
            Console.WriteLine("present");
        else if(b==0)
            Console.WriteLine("not present");
        else
            Console.WriteLine("Non Positive");
    }
}


class UserProgramCode
{
    public static int findExistence(int[] a)
    {
        int c = 0;
        int flag=0;
        for (int i = 1; i <= a[0]; i++)
        {
            if (a[i] >= 0)
            {
                if (a[i] == a[a[0] + 1])
                    c++;
            }
```

```
            else
            {
               flag = 1;
               break;
            }
         }
         if (flag == 0)
         {
            if (c > 0)
               return 1;
            else
               return 0;
         }
         else
            return -1;


      }
   }
}
```

102.GENERATE THE SERIES

Generate the series

Given a method taking an odd positive Integer number as input, write code to evaluate the following series:

1+3-5+7-9…+/-n.

Include a class UserProgramCode with a static method addSeries which accepts a positive integer . The return type of this method is an integer .

Create a class Program which would get the input as a positive integer and call the static method addSeries present in the UserProgramCode.

Input and Output Format:

Input consists of a positive integer n.

Output is a single integer .

Refer sample output for formatting specifications.

Sample Input 1:

9

Sample Output 1:

-3

Sample Input 2:

11

Sample Output 2:

8

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace level1_59
{
  class Program
  {
    static void Main(string[] args)
    {
      int n,x;
      n = Convert.ToInt32(Console.ReadLine());
      x = UserProgramCode.addSeries(n);
      Console.WriteLine(x);
```

```
        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace level1_59
{
    class UserProgramCode
    {
        public static int addSeries(int a)
        {
            int t = 0, k = 1;
            for (int i = 0; k <= a; i++)
            {
                if (i == 0)
                {
                    t = t + k;
                }
                else if (i == 1)
                {
                    t = t + k;
                }
                else if (i % 2 != 0)
                {
                    t = t + k;
                }
                else
                {
                    t = t - k;
```

```
            }

            k = k + 2;
          }
          return t;
      }
   }
}
```

103.CONCATENATE STRING

Concatenate String Write a program to concatenate two strings as per the following rules. Rules: 1.If the 2 strings are of same length, simply append them together and return the final string. 2.If the 2 given strings are of different length, remove starting characters from the longer string so that both strings are of same length and then append them together and return the final string. Include a class UserProgramCode with a static method concatstring that accepts a string and returns a string.

Create a Class Program which would be used to read 2 strings and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of two Strings. Output consists of a String.

Sample Input 1:

Hello hi

Sample Output 1:

Lohi

Sample Input 2:

cognizant coh

Sample Output 2:

Antcoh

Sample Input 3:

Hello hello

Sample Output 3:

Hellohello

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace progm54
{
   class Program
   {
      static void Main(string[] args)
      {
         string inp1 = Console.ReadLine();
         string inp2 = Console.ReadLine();
         string output = UserProgramCode.concatstring(inp1, inp2);
         Console.WriteLine(output);
      }
   }
class UserProgramCode
   {
      public static string concatstring(string inputstr1, string inputstr2)
      {
         int x = inputstr1.Length;
         int y = inputstr2.Length;
```

```
        string ans;
        if (x == y)
        {
            ans = inputstr1 + inputstr2;
        }
        else if (x > y)
        {
           int z = x - y;
           string inputstring1 = inputstr1.Remove(0, z);
           ans = inputstring1 + inputstr2;
        }
        else
        {
           int z = y - x;
           string inputstring2 = inputstr2.Remove(0, z);
           ans = inputstr1 + inputstring2;
        }
        return ans;

    }
  }
}
```

104.COUNT THE DIGITS

Count Digits Write a method to find number of digits present in the given string. Example: Input1: Hell00 ho9 are u Output1: 3 Include a class UserProgramCode with static method countDigits which accepts String value. The return type should be int.

Create a class Program which would get the input and call the static method countDigits present in the UserProgramCode.

The input String consists only alphabets , numeric values and whitespaces (blank spaces).Otherwise display as "Invalid Input". Input Output Format: Input consists of a String. Output consists of an integer which counts the number of digits in the given String.

Sample Input 1:

12345

Sample Output 1:

Number of digits present in given string are 5.

Sample Input 2:

hai12!

Sample Output 2:

Invalid Input

countdigits

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace count_the_digits
{
   class Program
   {
      static void Main(string[] args)
      {
         string str = Console.ReadLine();
         int op = UserProgramCode.CountDigits(str);
```

```
        if (op == -1)
            Console.WriteLine("Invalid Input");
        else
            Console.WriteLine("Number of digits present in the given string are {0}.",op);
        Console.ReadLine();
      }
   }
}


using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace count_the_digits
{
   class UserProgramCode
   {
      public static int CountDigits(string str)
      {
         int c = 0;
         for (int i = 0; i < str.Length; i++)
         {
            if (!char.IsLetterOrDigit(str[i])&&str[i]!=' ')
            {
               return -1;
            }
            if (char.IsDigit(str[i]))
            {
               c++;
            }
```

```
        }
        return c;
      }
   }
}
```

105.CONVERT FORMAT

Convert Format Write a program to convert a 10 digit positive number which is in the format XXX-XXX-XXXX to the format XX-XX-XXX-XXX. Include a class UserProgramCode with a static method convertFormat that accepts a string and returns a string. Create a Class Program which would be used to read the string call the static method present in UserProgramCode. Input and Output Format: Input consists of a String. Output consists of a string

Sample Input 1:
555-555-0000
Sample Output 1:
55-55-550-000
Sample Input 2:
000-000-0000
Sample Output 2:
00-00-000-000

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConvertFormat
{
   class Program
```

```
  {
    static void Main(string[] args)
    {
      string str = Console.ReadLine();
      string op = UserProgramCode.ConvertFormat(str);
      Console.WriteLine(op);
      Console.ReadLine();
    }
  }
}



using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConvertFormat
{
  class UserProgramCode
  {
    public static string ConvertFormat(string str)
    {
      int c = 0;
      StringBuilder sb = new StringBuilder();
      int len = str.Length;
      for (int i = 0; i < str.Length; i++)
      {
        if (str[i] == '-')
        {
          continue;
        }
```

```
            if (Char.IsDigit(str[i]))
            {
               sb.Append(str[i]);
               c++;
            }
            if (c % 2==0)
            {
               sb.Append("-");
            }
         }
         sb.Remove((sb.Length) - 1, 1);
         string final = sb.ToString();
         return final;
      }
   }
}
```

106.ADD & REVERSE

Given an int array and a number 'k' as input, write a program to add all the elements in the array greater than the given number 'k'. Finally reverse the digits of the obtained sum and print it.

Include a class UserProgramCode with a static method “addAndReverse” that accepts 2 arguments and returns an integer.The first argument corresponds to the integer array and the second argument corresponds to the number k.

Create a class Program which would get the required input and call the static method addAndReverse present in the UserProgramCode.

Example:
Input Array = {10,15,20,25,30,100}
Number = 15

sum = 20 + 25 + 30 + 100 = 175

output = 571

Input and Output Format:

The first line of the input consists of an integer that corresponds to the number of elements in the array.

The next n lines of the input consists of integers that correspond to the elements in the array.

The last line of the input consists of an integer that corresponds to the number, k.

Output consists of a single integer.

Sample Input

6

10

15

20

25

30

100

15

Sample Output

571

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Linq;

public class Program
{
   public static void Main(string[] args)
```

```
    {
        int n = Convert.ToInt32(Console.ReadLine());
        int[] ar = new int[n];
        for (int i = 0; i < n; i++)
        {
            ar[i] = Convert.ToInt32(Console.ReadLine());
        }
        int k = Convert.ToInt32(Console.ReadLine());
        int res = UserMainCode.AddReverse(ar, k);
        Console.WriteLine(res);

            Console.Read();
    }
}



using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
class UserMainCode
{
    public static int AddReverse(int[] ar,int k)
    {
        int sum = 0,rem,rev=0;
        for (int i = 0; i < ar.Length; i++)
        {
            if (ar[i] > k)
            {
                sum = sum + ar[i];
            }
        }
```

```
        while (sum > 0)
        {
            rem = sum % 10;
            rev = rev * 10 + rem;
            sum = sum / 10;

        }
        return rev;
    }

}
```

107.FIXED POINT

Given an input array of n distinct integers sorted in ascending order, write a program that finds a Fixed Point in the array . Fixed Point in an array is an index i such that arr[i] is equal to i. Business Rules : 1. If any of the given inputs contain any negative number, then print -1. 2. If there are no fixed point values found in the input array, then print -2. 3. If there are less than 2 elements or more than 10 elements in the input array, then print -3.

Assume that there will be a maximum of 1 fixed point in the input array.

Create a class named UserProgramCode that has the following static method

public static int findFixedpoint(int[] input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array 1.

The next 'n' lines of input consist of elements in the input array 1.

Refer business rules and sample output for formatting specifications.

Sample Input 1 :

6 1 4 45 3 0 19

Sample Output1 :

3

Sample Input 2 :

5 1 10 5 2 -7

Sample Output 2 :

-1

FIXED POINT

```
using System;


class Program

{

 public static void Main( string[] args )

 {
     int n = Convert.ToInt32(Console.ReadLine());
     int[] array = new int[n];
     for (int i = 0; i < n; i++)

array[i] = Convert.ToInt32(Console.ReadLine());

int result=UserProgramCode.findFixedpoint(array);


  Console.WriteLine(result);

 Console.Read();
```

```
 }

 }
```

```
using System;

class UserProgramCode
{
   public static int findFixedpoint(int[] input)
   {
      Array.Sort(input);
      int res=0;
      int count=0;
      foreach (int k in input)
      {
         //Console.WriteLine(k);
         if (k < 0)
         {
            count++;
            return -1;
         }
      }
      if (input.Length < 2 || input.Length > 10)
      {
         count++;
         return -3;
      }
      if (count == 0)
      {
         int i = 1;
         foreach(int k in input)
```

```
        {
            if (k==i)
                res = i-1;
            i++;
        }
        if (res == 0)
            res = -2;
    }
    return res;
  }
}
```

108.ONLINE SALES

An online shopping portal announced a big bang sale,the discounts apply based on purchased time. The discount sales would start from 10 am and will end by 6pm. The discount is not applicable when the products are purchased outside the window time. A) If the product is bought between 10am - 11am,then customer gets 50% off. B) If the product is bought after 11am but within 12pm,then customer gets 40% off. C) If the product is bought after 12pm but within 4pm,then customer gets 30% off. D) If the product is bought after 4pm,only 25% off. The actual price and the time of buying the product are given as input1 and input2 respectively. The time is given as integer in the format as hhmm where hh refers to the hours in 24 hrs time format and mm refers to the minutes . Write a program to calculate the discounted price of the product and print the output in the following format. The actual price of the product is Rs XXX and you have bought it for Rs YYY. You Save Rs ZZZ. Here XXX refers to the actual price of the product, YYY refers to the price after the discount is applied, and ZZZ refers to the difference in between the actual and the discounted price if any. Business rules: 1) If the actual price is zero or less than zero, Print 'Invalid Price Amount' 2) If the product is bought outside the window time , print the output in the following format : The price of the product is Rs XXX and discounts are not applicable.

Example 1: input1 : 20000 input2 : 1538(3 PM 38 mins) output : The actual price of the product is Rs 20000 and you have bought it for Rs 14000. You Save Rs 6000. Example 2: input1 :-40 input2 : 1038(10 AM 38 mins) output1 : Invalid Price Amount

Create a class named UserProgramCode that has the following static method

public static void onlineSalesAmount(int input1,int input2)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer that corresponds to the cost.

The second line of the input consists of an integer that corresponds to the time.

Refer business rules and sample output for output format.

Display the price after discount as an integer only.

Sample Input 1 :

20000 1538

Sample Output 1 :

The actual price of the product is Rs 20000 and you have bought it for Rs 14000. You Save Rs 6000.

Sample Input 2 :

-40 1038

Sample Output 2 :

Invalid Price Amount

```
using System;
using System.Text.RegularExpressions;
namespace code1
{
  class Program
{
  static void Main(String[] args)
{
  int input1;
  int input2;
```

```
input1=int.Parse(Console.ReadLine());
input2 = int.Parse(Console.ReadLine());
  UserMainCode.onlineSalesAmount(input1, input2);




      }
}

}
```

```
using System;


public class UserMainCode
{




   public static void onlineSalesAmount(int input1, int input2)
{
//Console.WriteLine(input1);
  if (input1 > 0)
{
  if (input2 > 1000 && input2 <= 1100)
{
  Console.WriteLine("The actual price of the product is Rs " + input1 + " and you have bought it for Rs " +
(input1 - (input1 * 0.5)) + ". You Save Rs " + (input1 * 0.5) + ".");
}
```

```
  else if (input2 > 1100 && input2 <= 1200)
{
  Console.WriteLine("The actual price of the product is Rs " + input1 + " and you have bought it for Rs " +
(input1 -(input1 * 0.4)) + ". You Save Rs " + (input1 * 0.4) + ".");
}
  else if (input2 > 1200 && input2 <= 1600)
{
  Console.WriteLine("The actual price of the product is Rs " + input1 + " and you have bought it for Rs " +
(input1 -(input1 * 0.3)) + ". You Save Rs " + (input1 * 0.3) + ".");
}
  else if (input2 > 1600 && input2 <= 1800)
{
  Console.WriteLine("The actual price of the product is Rs " + input1 + " and you have bought it for Rs " +
(input1 -(input1 * 0.25)) + ". You Save Rs " + (input1 * 0.25) + ".");
   }
  else
  Console.WriteLine("The price of the product is Rs "+input1+" and discounts are not applicable.");
}
  else
  Console.WriteLine("Invalid Amount");
  }
}
```

109.FIND THE SHORTEST STRING

Write a program that reads an Integer (size of the String List), a String List and a character. Find the shortest string from the list that starts with the character. (case sensitive). Assume there will be only one string.

Include a class UserProgramCode with static method GetshortestString() that accepts a string list and a character . The return type is String.

Create a class Program which would get the input and call the static method GetshortestString(List<string> input1, char input2)) present in the UserProgramCode.

Input and Output Format :

The first input corresponds to the list size.

The next input corresponds to the elements in the list which is a string.

The third input is a character.

Output is String.

In GetshortestString()

1. Only alphabets should be given in the List else return "Non Alphabets Exists". 2. If there is no match found in input then return "No String Found".

3. Otherwise return the appropriate result. In Program Display the result which is return by GetshortestString()

Sample Input 1:

4 read elegant Edit even e

Sample Output 1:

even

Sample Input 2:

2 qwerty abcdef e

Sample Output 2:

No String Found

Sample Input 3:

4 re@d el3gant Edit even e

Sample Output 3:

Non Alphabets Exists

NO ANSWER FOUND

110.SORT STRING

Given a string list as input{StringOne, StringTwo, StringThree,...}, Write a program to sort each character in the individual string in ascending order of alphabets {eginnOrSt, ginorStTw, eeghinrrStT,..}, then remove repetitive characters irrespective of case {eginOrSt, ginorStw, eghinrST,..}, then sort the resultant list in ascending order and print the array of string in lowercase{eghinrst,eginorst, ginorstw,..}. Ignore the case sensitivity of given input. Business Rule: 1. If any of the elements in the given input

contains any special characters or numbers, print 'Invalid Input'. 2. If there are duplicate elements in the given input,, remove the duplicates and follow all other steps. Example1 : input : 4 Ccaat rat dog cow output : act art cow dgo Steps: 1. Sort each character of each string element: {aacCt,art,dgo,cow} 2. Remove repetitive characters irrespective of the case: {aCt,art,dgo,cow} 3. Now sort each string element and arrange it in ascending order in lowercase: {act,art,cow,dgo} Example 2 : input 4 cat rat dog1 cow$ output Invalid Input Example 3 : input 5 cat rat cat cow rat output act art cow

Create a class named UserProgramCode that has the following static method

public static List<string> sortString(List<string> arr) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.

The next 'n' lines of input consist of elements in the input array.

Refer business rules and sample output for formatting specifications.

Sample Input :

4 Ccaat rat dog cow

Sample Output :

act art cow dgo

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace sort_String
{
   class UserprogramCode
   {
      public static List<string> sortString(List<string> arr)
      {
         List<StringBuilder> list1 = new List<StringBuilder>();
```

```
char temp;
foreach (var st in arr)
{
   string str = st.ToLower();
   StringBuilder sb = new StringBuilder(str);

   for (int i = 0; i < sb.Length - 1; i++)
   {
      for (int j = i + 1; j < sb.Length; j++)
      {
         if (sb[i] > sb[j])
         {
            temp = sb[i];
            sb[i] = sb[j];
            sb[j] = temp;
         }
      }


   }
   list1.Add(sb);

}
foreach (var sb in list1)
{
   for (int i = 0; i < sb.Length - 1; i++)
   {

      if (sb[i].ToString().ToLower() == sb[i + 1].ToString().ToLower())
      {
         sb.Remove(i, 1);
      }
```

```
            }

        }
        StringBuilder tempo;

        for (int i = 0; i < list1.Count-1; i++)
        {
            for (int j = i+1; j < list1.Count; j++)
            {
                if (list1[i][0] > list1[j][0])
                {
                    tempo = list1[i];
                    list1[i] = list1[j];
                    list1[j] = tempo;
                }
            }
        }
        List<string> final = new List<string>();
        foreach (var str in list1)
        {
            final.Add(str.ToString());
        }
        return final;
    }
  }
}


using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace sort_String
{
   class Program
   {
      static void Main(string[] args)
      {
         int n = int.Parse(Console.ReadLine());
         List<string> list = new List<string>();
         for (int i = 0; i < n; i++)
         {
            list.Add(Console.ReadLine());
         }
         List < string > op= UserprogramCode.sortString(list);
         foreach (var item in op)
         {
            Console.WriteLine(item);
         }


      }
   }
}
```

111.CALCULATE CHARGE

A parking garage charges a Rs.20 minimum fee to park for up to three hours. The garage charges an additional Rs. 5 per hour for each hour or part thereof in excess of three hours. The maximum charge for any given 24-hour period is Rs.100. Assume that no car parks for longer than 24 hours at a time.

Write a program which accepts two DateTime inputs as string datatype. Convert the inputs to DateTime

DataType and calculate the parking charge for the vehicle.

Validations:

1. DateTime String format is “yyyy-MM-dd:HH:mm:ss” eg: 2009-10-21:14:35:45 .

Return -1 as error code for other formats. The first parameter in the method will refer to the checkinDate and the second would refer to checkoutDate.

2. CheckoutDateTime should be greater than check in time. Return -2 if that is not the case.

3. If the duration exceeds 24 hrs return -3 as error code.

Include a class UserProgramCode with a static method calculateCharge which accepts two Strings. The return type (Integer) should return the parking charge. Also follow the validations.

Create a Class Program which would be used to accept two Strings, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of two Strings, the first String corresponds to the CheckinDateTime and the second String corresponds to the CheckoutDateTime.

Output consists of an Integer (the parking charges) or, a String “Invalid Date format” if -1 is returned, “CheckoutDateTime is less than CheckinDateTime” if -2 is returned, “Duration exceeds 24 hrs” if -3 is returned.

Refer sample output for formatting specifications.

Sample Input 1:

2009-10-21:14:35:45

2009-10-21:16:35:45

Sample Output 1:

20

Sample Input 2:

2009-10-211:14:35:45

2009-10-21:16:35:45

Sample Output 2:

Invalid Date format

Sample Input 3:

2009-10-21:14:35:45

2009-10-21:10:35:45

Sample Output 3:

CheckoutDateTime is less than CheckinDateTime

Sample Input 4:

2009-10-20:14:35:45

2009-10-21:16:35:45

Sample Output 4:

Duration exceeds 24 hrs

```
using System;

class Program
{
 public static void Main( string[] args )
 {
    string date1 = Console.ReadLine();
    string date2 = Console.ReadLine();

    int result = UserProgramCode.getDateDifference(date1, date2);
    Console.WriteLine(result);
    Console.ReadLine();
 }
}
using System;
using System.Text;
using System.Linq;
```

```
using System.Text.RegularExpressions;

class UserProgramCode
{
  public static int getDateDifference(string date1,string date2)
  {
     int result=0;
     int time = 0;
     DateTime dt1;
     DateTime dt2;
     bool res1 = DateTime.TryParseExact(date1, "yyyy-MM-dd:HH:mm:ss", null,
System.Globalization.DateTimeStyles.None, out dt1);
     bool res2 = DateTime.TryParseExact(date2, "yyyy-MM-dd:HH:mm:ss", null,
System.Globalization.DateTimeStyles.None, out dt2);
     if (res1 == true && res2 == true)
     {
         time = (int)dt2.Subtract(dt1).TotalHours;
        if (time < 0)
        {
           result= -2;
        }
        else if (time > 24)
        {
           result= -3;
        }
        else
        {
           if (time <= 3)
           {
              result = 20;
           }
           else
```

```
        {
           result = time * 5;
        }

        if (result > 100)
        {
           result = 100;
        }
      }
   }
   else
   {
      result = -1;
   }
      return result;
 }
}
```

112.SORT ARRAY ELEMENT

Sort Array Element Write a method to sort input string array by the length of each element. If word length is same then sort thiese two words in ascending order without considering length.

Include a class UserProgramCode with a static method sortArrayElement which accepts a string array as input. The return type of a method is string array. If input contains any special characters then add '-1' into the list.

Create a class Program which would get the input and call the static method sortArrayElement present in the UserProgramCode.

Input and Output format : The first line of the input consists of an integer that corresponds to the number of elements in the string array. The nexr 'n' lines consist of string inputs. Output consists of array which contains sorted elements or "-1".

Sample Input 1 :

3 Greenapple Litchi Mango

Sample Output 1 : Mango Litchi Greenapple

Sample Input 2 : 2 one #two

Sample Output 2 : -1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;

namespace ConsoleApplication18
{
   class Program
   {
      static void Main(string[] args)
      {
         int n = Convert.ToInt32(Console.ReadLine());
         string[] a = new string[n];
         string[] b;
         for (int i = 0; i < a.Length; i++)
            a[i] = Console.ReadLine();
         b = userProgramCode.sortArrayElement(a);
         foreach(string c in b)
            Console.WriteLine(c);
         Console.ReadLine();
      }
   }
   class userProgramCode
   {
      public static string[] sortArrayElement(string[] a)
      {
         string[] b=new string[1];
         for (int i = 0; i < a.Length; i++)
```

```
        {
            for (int j = 0; j < a[i].Length; j++)
            {
                if (!char.IsLetterOrDigit(a[i][j]))
                {
                    b[0] = "-1";
                    return b;
                }
            }
        }
        Array.Sort(a, StringComparer.Ordinal);
Array.Reverse(a);
        return a;
      }
   }
}
```

113. REMOVE TENS

Given an input array, write a program to remove all Tens present in the array and shift the other elements towards left and fill the trailing empty positions by 0 so that the modified array is of the same length of the given array a

Include a class UserProgramCode with static method removeTens that accepts an integer array and its size and returns the modified array.

Create a class Program which would get the input array and call the static method removeTens present in the UserProgramCode.

Input and Output Format:

Input consists of an n+1 integers. The 1st integer corresponds to n, the size of the array. The remaining n integers correspond to the element in the array.

Output is the modified integer array.

Sample Input 1:

5

1 10 20 10 2

Sample Output 1:

1 20 2 0 0

Sample Input 2:

2 1 2

Sample Output 2:

1 2

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace tens
{
   class Program
   {
      static void Main(string[] args)
      {
         int n = Convert.ToInt32(Console.ReadLine());
         int[] a = new int[n];
         for (int i = 0; i < n; i++)
         {
            a[i] = Convert.ToInt32(Console.ReadLine());
         }
         int[] op = new int[100];
         op = UserProgramCode.removeTens(a, n);
```

```
        foreach (int item in op)
            Console.WriteLine(item);
        Console.ReadLine();

    }
  }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace tens
{
  class UserProgramCode
  {
    public static int[] removeTens(int[] arr, int size)
    {
      int count = 0, k = 0;
      int j;
      int[] output = new int[size];

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

        if (arr[i] != 10)
        {
          output[k] = arr[i];
          k++;
        }
        else
        {
```

```
            count++;

          }

        }

        for (j = 0; j < count; j++)
        {

          output[k] = 0;
          k++;
        }
        return output;
      }


  }
}
```

114. SUM OF SQUARES & CUBES OF ELEMENTS IN THE ARRAY

Write a program to get an int array as input and identify even and odd numbers. If number is odd get cube of it, if number is even get square of it. Finally add all cubes and squares together and return it as output.

Include a class UserProgramCode with a static method addEvenOdd which accepts an integer array as input and returns an integer.

The method returns an integer which is the sum of cubes of all odd numbers and squares of all even numbers in the array.

Create a class Program which would get the input and call the static method addEvenOdd present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer n, that corresponds to the number of elements in the array. The next 'n' lines of input consists of the elements in the array.

Output is an integer that corresponds to the required sum.

Refer sample output for formatting specifications.

Sample Input 1:

5

2

6

3

4

5

Sample Output 1:

208

63

```
class Program
{
   static void Main(string[] args)
   {
      int n;
      n = Convert.ToInt32(Console.ReadLine());
      int[] a=new int[n];
      for (int i = 0; i < n; i++)
      {
         a[i] = Convert.ToInt32(Console.ReadLine());
      }

      userprogramcode obj = new userprogramcode();
```

```
        n=obj.addEvenOdd(a);
        Console.WriteLine(n);

    }
}
public class userprogramcode
{
    public int addEvenOdd(int[] a)
    {
        int sum=0;
        foreach (var n in a)
        {
            if (n % 2 == 0)
                sum += Convert.ToInt32(Math.Pow(n, 2));
            else
                sum += Convert.ToInt32(Math.Pow(n, 3));

        }
        return sum;
    }
}
```

115**. BONUS CALCULATION**

Write a program to calculate the bonus of the employee given the basic salary of the employee. The bonus will be calculated based on the below category. If Basic Salary>15000 and less than 20001 calculate bonus as 17% of basic+1500 If Basic Salary>10000 and less than 15001 calculate bonus as 15% of basic+1200 If Basic Salary<10001 calculate bonus as 12% of basic+1000 for rest calculate bonus as 8%of basic+500 Business rule: 1) If the salary given is a negative number, then print -1. 2) If the salary given is more than 1000000, then print -2 . 3) All the test cases has the calculated bonus as integer value only. Create a class named UserProgramCode that has the following static method

public static int calculateBonus(int input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of an integer that corresponds to the salary.

Output is an integer.

Refer sample output and business rule for output formatting specifications.

Sample Input 1 :

10000

Sample Output1 :

2200

Sample Input 2 :

2000000

Sample Output 2 :

-2

calculate bonus

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace bonus
{
   class Program
   {
      static void Main(string[] args)
      {
         int basic = int.Parse(Console.ReadLine());
         int op = UserProgramCode.CalculateBonus(basic);

         Console.WriteLine(op);
```

```
        Console.ReadLine();
      }
   }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace bonus
{
   class UserProgramCode
   {
      public static int CalculateBonus(int basic)
      {
         int bonus = 0;
         if (basic < 0)
            return -1;
         if (basic > 1000000)
            return -2;
         if (basic < 20001 && basic > 15000)
         {
            bonus=Convert.ToInt32(basic*0.17)+1500;
         }
         if(basic>10000&&basic<15001)
            bonus = Convert.ToInt32(basic * 0.15) + 1200;
         if(basic<10001)
            bonus = Convert.ToInt32(basic * 0.12) + 1000;
         else
```

```
            bonus = Convert.ToInt32(basic * 0.08) + 500;
          return bonus;
       }
    }
}
```

116.**INITIAL FORMAT**

Write a program to input a person's name in the format "FirstName LastName" and return the person name in the following format - "LastName, InitialOfFirstName".

Include a class UserProgramCode with a static method nameFormatter which accepts a string. The return type (string) should return the expected format.

Create a Class Program which would be used to accept Input String and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a string that corresponds to a Person's name.

Output consists of a string(person's name in expected format).

Refer sample output for formatting specifications.

Sample Input :

Jessica Miller

Sample Output:

Miller, J

```
class Program
{
   static void Main(string[] args)
   {
```

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

        userprogramcode obj = new userprogramcode();
        Console.WriteLine(obj.nameFormatter(s));

    }
}
public class userprogramcode
{
    public string nameFormatter(string s)
    {
        string[] str;
        str = s.Split(' ');
        s = str[1] + ", " + str[0][0];
        return s;
    }
}
```

117.EXTRACT MAX SUBSTRING

Write method to get a the substring with maximum number of characters for given string input (input1) separated by given delimeter (input2). If two or more substrings have maximum number of characters return the substring which appears first in the alphabetical order. Example : Input1 = "delhi-pune-patna" Input2 = "-" Output1 = "delhi"

Create a class named UserProgramCode that has the following static method

public static string extractMax(string input1, string input2)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of 2 strings.

Output is a string.

Sample Input : delhi-pune-patna

-

Sample Output :

delhi

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Extract_max_SubString1
{
   class UserProgramCode
   {
      public static string extractMax(string str, string limit)
      {
         List<string> list = new List<string>();
         string[] arr = str.Split(char.Parse(limit));
         int max = 0;
         for (int i = 0; i < arr.Length; i++)
```

```
        {
            if (arr[i].Length > max)
                max = arr[i].Length;
        }
        for (int i = 0; i < arr.Length; i++)
        {
            if (arr[i].Length == max)
                list.Add(arr[i]);
        }

        list.Sort();
        return list[0];

    }
  }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
namespace Extract_max_SubString1
{
   class Program
   {
      static void Main(string[] args)
      {
         string str = Console.ReadLine();
         string str2 = Console.ReadLine();
         string op = UserProgramCode.extractMax(str, str2);
         Console.WriteLine(op);
         Console.ReadLine();
 }
   }
}
```

118.FIND LOWEST

Write a program to find the lowest number in an integer array.
Print the lowest number.
Only positive number should be given as input in an array. Else print “Negative numbers present”.

Include a class UserProgramCode with a static method findLowest which accepts an Integer array. The return type (Integer) should return the lowest number. If negative numbers are present in the array, then return -1.

Create a Class Program which would be used to accept an Integer and an Integer array, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of n+1 Integers, where the first number corresponds the size of the array, followed by the array elements.

Output consists of an Integer, the lowest number, or a String “Negative numbers present” if a negative number is present in the array.

Refer sample output for formatting specifications.

Sample Input 1:

4

2

3

1

8

Sample Output 1:

1

Sample Input 2:

4

2

3

-1

8

Sample Output 2:

Negative numbers present

45. Find Lowest

```
using System;
using System.Collections.Generic;
```

```
using System.Linq;
using System.Text;

namespace Workout45
{
   class UserProgramCode
   {
      public static int findLowest(int[] a)
      {
         int i, j, t, n;
         n = a.Length;
         t = a[0];
         for (i = 0; i < n; i++)
         {
            for (j = i + 1; j < n+1; j++)
            {
               if (a[i] < 0)
               {
                  t = -1;
                return t;
               }
               else if (t > a[i])
               {
                  t = a[i];
                  a[i] = a[j];
                  a[j] = t;
               }
            }
         }
         return t;
      }
   }
```

```
  class Program
  {
     static void Main(string[] args)
     {
        //UserProgramCode u = new UserProgramCode();
        int i,n,s;
        n = int.Parse(Console.ReadLine());
        int[] a = new int[n];
        for (i = 0; i < n;i++ )
           a[i] = int.Parse(Console.ReadLine());
        s = UserProgramCode.findLowest(a);
        if(s>=0)
           Console.WriteLine(s);
        else if (s < 0)
           Console.WriteLine("Negative Numbers present");
        else { }
     }
  }
}
```

119.CHECK SUM

Write program to read a positive int as input and to calculate the sum of the odd digits in the given number. If the sum is odd print “Odd” else print “Even”.

Example:
input = 56895
Sum = 5 + 9 + 5 = 19 (odd)
output = Odd

Include a class UserProgramCode with a static method checkSum which accepts a positive Integer. The

return type (Integer) should return 1 if the sum is odd, else return -1.

Create a Class Program which would be used to accept a positive Integer, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of an Integer.

Output consists of a String “Odd” if the sum is odd, else print “Even”.

Refer sample output for formatting specifications.

Sample Input:

56895

Sample Output

Odd

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace question36
{
   class Program
   {
      static void Main(string[] args)
      {
         int n, c;
         n = Convert.ToInt32(Console.ReadLine());
         c = UserProgramCode.checkSum(n);
         if (c == 1)
            Console.WriteLine("Odd");
         else Console.WriteLine("Even");
      }
```

```
  }


  class UserProgramCode
  {
    public static int checkSum(int a)
    {
       int rem, sum = 0;
       while (a > 0)
       {

          rem = a % 10;
          if (rem % 2 == 1)
          { sum = sum + rem; }
          a = a / 10;
       }

       if (sum % 2 == 1)
          return (1);
       else return (-1);

    }

  }

}
```

120. CALCULATE DISCOUNT

Calculate Discount Given two input integer arrays input1 and input2 which contains the details of users booking cars online in the format of key value pairs. input1 (Key,Value) = (UserId1,Bookingmonth1,UserId2,Bookingmonth2,...so on) input2 (Key,Value) = (UserId1,BookingAmount1,UserId2,BookingAmount2,...so on) Write a Program to calculate the discount

amount based on the Booking amount and Booking month for every UserId by considering the respective discount rate offered to them and store the result in an output integer array .

The discount amount should be calculated using the following specifications ---------- ---------- Discount Rate Booking Amount ---------- ---------- - | >=500000 | >=100000 and < 500000 ---------- ---------- Discount Rate for Jan-April | 25 | 15 Discount Rate for May-August | 20 | 10 Discount Rate for Sep-December | 15 | 5 ---------- ---------- Output should be in the below format: output1 (Key,Value) = (UserId1,DiscountAmount1,UserId2,DiscountAmount2,...so on) Note: 1)Formula: Discount Amount = Booking Amount * (Discount Rate/100) BUSINESS RULE: 1.If Booking Amount is less than 100000, then discount amount should be zero. 2.If any of the booking month is invalid, then print -1. 3.If any of the booking amount is negative, then print -2. 4.If any of the Booking Month or Booking Amount is missing for any UserId, then print -3 . Create a class named UserProgramCode that has the following static method

public static int[] calcDiscount(int[] input1, int[] input2)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, m that corresponds to the number of elements in the input array 1 .

The next 'm' lines of input correspond to elements in the input array 1.

The next line of the input consists of an integer, n that corresponds to the number of elements in the input array 2 .

The next 'n' lines of input correspond to elements in the input array 2.

Refer business rules and sample output for formatting specifications. Sample Input 1 :

8 1010 11 1011 02 1012 07 1013 09 8 1010 700000 1011 300000 1012 150000 1013 100000

Sample Output 1 : 1010 105000 1011 45000 1012 15000 1013 5000

Sample Input 2 : 8 1010 11 1011 02 1012 07 1013 09 6 1010 700000 1011 300000 1012 100000

Sample Output 2: -3

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication7
{
   class Program
   {
      static void Main(string[] args)
      {
         int m, n,i=0;
         m = Convert.ToInt32(Console.ReadLine());
         int[] a = new int[m];
         int[] output = new int[m];
         for (i = 0; i < m; i++)
         {
            a[i] = Convert.ToInt32(Console.ReadLine());
         }
         n = Convert.ToInt32(Console.ReadLine());
         int[] b = new int[n];
         for (i = 0; i < n; i++)
         {
            b[i] = Convert.ToInt32(Console.ReadLine());
         }
         output = UserProgramCode.calcDiscount(a, b);
         if (output[0] < 0)
         {
            Console.WriteLine(output[0]);
         }
```

```
        else
        {
           for (i = 0; i < m; i++)
           {

              Console.WriteLine(output[i]);
           }
        }
        Console.Read();

     }
  }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication7
{
   class UserProgramCode
   {
      public static int[] calcDiscount(int[] input1, int[] input2)
      {
         int len1 = 0, len2 = 0,i=0;
         int discount=0;
         len1 = input1.Length;
         len2 = input2.Length;
```

```
int[] c = new int[len1];
if (len1 != len2)
{
   c[0] = -3;
   return c;
}
else
{
   for (i = 1; i < len1; i = i + 2)
   {
      if (!(input1[i] >= 1 && input1[i] <= 12))
      {
         c[0] = -1;
         return c;
      }
   }
   for (i = 1; i < len2; i = i + 2)
   {
      if ((input2[i] < 0))
      {
         c[0] = -2;
         return c;
      }
   }
   for (i = 1; i < len1; i=i+2)
   {
      if (input2[i] < 100000)
      {
         discount = 0;
      }
      if (input2[i] >= 500000)
      {
```

```
        if (input1[i] >= 1 && input1[i] <= 4)
        {
            discount = input2[i] * 25;
        }
        if (input1[i] >= 5 && input1[i] <= 8)
        {
            discount = input2[i] * 20;
        }
        if (input1[i] >= 9 && input1[i] <= 12)
        {
            discount = input2[i] * 15;
        }
    }
    if ((input2[i] >= 100000) && (input2[i]<500000))
    {
        if (input1[i] >= 1 && input1[i] <= 4)
        {
            discount = input2[i] * 15;
        }
        if (input1[i] >= 5 && input1[i] <= 8)
        {
            discount = input2[i] * 10;
        }
        if (input1[i] >= 9 && input1[i] <= 12)
        {
            discount = input2[i] * 5;
        }
    }

    c[i] = discount/100;
}
for (i = 0; i < len1; i=i+2)
```

```
            {
               c[i] = input1[i];
            }
            return c;
         }
      }
   }
}
```

121-150

121.calculate Discount

Calculate Discount Given two input integer arrays input1 and input2 which contains the details of users booking cars online in the format of key value pairs. input1 (Key,Value) = (UserId1,Bookingmonth1,UserId2,Bookingmonth2,...so on) input2 (Key,Value) = (UserId1,BookingAmount1,UserId2,BookingAmount2,...so on) Write a Program to calculate the discount amount based on the Booking amount and Booking month for every UserId by considering the respective discount rate offered to them and store the result in an output integer array .

The discount amount should be calculated using the following specifications ---------- ---------- Discount Rate Booking Amount ---------- ---------- ---------- - | >=500000 | >=100000 and < 500000 ---------- ---------- Discount Rate for Jan-April | 25 | 15 Discount Rate for May-August | 20 | 10 Discount Rate for Sep-December | 15 | 5 ---------- ---------- Output should be in the below format: output1 (Key,Value) = (UserId1,DiscountAmount1,UserId2,DiscountAmount2,...so on) Note: 1)Formula: Discount Amount = Booking Amount * (Discount Rate/100) BUSINESS RULE: 1.If Booking Amount is less than 100000, then discount amount should be zero. 2.If any of the booking month is invalid, then print -1. 3.If any of the

booking amount is negative, then print -2. 4.If any of the Booking Month or Booking Amount is missing for any UserId, then print -3 . Create a class named UserProgramCode that has the following static method

public static int[] calcDiscount(int[] input1, int[] input2)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, m that corresponds to the number of elements in the input array 1 .

The next 'm' lines of input correspond to elements in the input array 1.

The next line of the input consists of an integer, n that corresponds to the number of elements in the input array 2 .

The next 'n' lines of input correspond to elements in the input array 2.

Refer business rules and sample output for formatting specifications.

Sample Input 1 :

8

1010 11 1011 02 1012 07 1013 09 8 1010 700000 1011 300000 1012 150000 1013 100000

Sample Output 1 : 1010 105000 1011 45000 1012 15000 1013 5000

Sample Input 2 : 8 1010 11 1011 02 1012 07 1013 09 6 1010 700000 1011 300000 1012 100000

Sample Output 2: -3

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;
```

```
namespace ConsoleApplication7
{
   class Program
   {
      static void Main(string[] args)
      {
         int m, n,i=0;
         m = Convert.ToInt32(Console.ReadLine());
         int[] a = new int[m];
         int[] output = new int[m];
         for (i = 0; i < m; i++)
         {
            a[i] = Convert.ToInt32(Console.ReadLine());
         }
         n = Convert.ToInt32(Console.ReadLine());
         int[] b = new int[n];
         for (i = 0; i < n; i++)
         {
            b[i] = Convert.ToInt32(Console.ReadLine());
         }
         output = UserProgramCode.calcDiscount(a, b);
         if (output[0] < 0)
         {
```

```
            Console.WriteLine(output[0]);
        }
        else
        {
            for (i = 0; i < m; i++)
            {

                Console.WriteLine(output[i]);
            }
        }
        Console.Read();

    }
  }
}
```

122.string processing 1

String Processing I

Given a string input input1, write a program to fetch the last n characters from input1 and repeat them after input1 the same number of times as given in the second integer input input2. Business Rules : 1. If the input1 contains any number, print -1. 2. If the input1 contains any special characters, print -2. 3. If the input1 string contains less than input2 number of characters,then print -3.

Create a class named UserProgramCode that has the following static method

public static string getString(string input1,int input2)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of a string and the second line of the input consists of an integer.

Refer sample output for formatting specifications.

Sample Input 1: Cognizant 3

Sample Output 1:

Cognizantantantantant Sample Input 2:

Teach123er 4 Sample Output 2:

-1

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;



namespace ConsoleApplication13

{

   class userprogramcode

   {

      public static string getString(string ip1, int ip2)

      {

         string s,final=ip1;
```

```
        char s1;
        for (int i = 0; i < ip1.Length; i++)
        {
           s1 = ip1[i];



            if (char.IsNumber(s1))
              return "-1";
           else if (!char.IsLetter(s1))
              return "-2";
        }
        s = ip1.Substring(ip1.Length - (ip2));
        int j = ip2;
        while (j > 0)
        {
           final = final + s;
           j--;
        }
        return final;
    }
}
class Program
{
```

```
    static void Main(string[] args)
    {
        String x,y;
        int k;
        x = Console.ReadLine();
        k = Convert.ToInt32(Console.ReadLine());
        y = userprogramcode.getString(x,k);
        Console.WriteLine(y);
    }
  }
}
```

123. Duplicate Date Elements

Write a program to eliminate duplicate date elements in an input String Array. Print the resultant String array/list in dd/MM/yyyy format. Business Rule: 1. Input Date will be of the form: dd-MM-yyyy or dd/MM/yyyy or dd.MM.yyyy or dd-Month-yyyy. 2. If any date is invalid, print 'Invalid Date'. Create a class named UserProgramCode that has the following static method
public static List<string> removeDuplicateDate(string[] input1) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:
The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.
The next 'n' lines of input consist of elements in the input array.
Refer business rules and sample output for formatting specifications. Sample Input 1 : 5 20/09/2014 30-03-2015 13.06.2012 20-09-2014 20-September-2014
Sample Output 1 : 20/09/2014 30/03/2015 13/06/2012 Sample Input 2 :

5 20/09/2014 30-03-2015 13.13.2012 20-09-2014 20-September-2014

Sample Output 2 : Invalid Date

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace duplicatedate
{
   class Program
   {
      static void Main(string[] args)
      {
         int n = Convert.ToInt32(Console.ReadLine());
         List<string> d = new List<string>(n);
         List<string> e = new List<string>(n);
         for (int i = 0; i < n; i++)
            d.Add(Console.ReadLine());
         e = pp.fun(d);
         foreach (string f in e)
            Console.WriteLine(f);
         Console.ReadLine();
```

```
    }
}

class pp
{
    public static List<string> fun(List<string> d)
    {
        DateTime dt1;
        List<string> f = new List<string>();
         List<string> z= new List<string>();
        foreach (string a in d)
        {
            bool i = DateTime.TryParseExact(a, "yyyy-MM-dd", null,
System.Globalization.DateTimeStyles.None, out dt1);
            if (i == true)
            {
                string k = dt1.ToString("yyyy-MM-dd");
                f.Add(k);
            }
             if (i == false)
            {
```

```
            z.Add("-1");
            return z;
          }
        }
        f=  f.Distinct().ToList();
         return f;
         }
    }
}
```

124.CheckCharacters

Given a method with a string input, write code to test if first and last characters are same. Incase they are same return 1 else return -1 as output. Note - Consider case.

Example:
Input = ""the picture was great""
first character = 't'
last character = 't'
Output = 1

Include a class UserProgramCode with static method checkCharacters that accepts a string and returns an integer. Create a class Program which would get the input and call the static method checkCharacters present in the UserProgramCode.
Input and Output Format: Input is a String - a sentence Output is a String --- “The characters are same” or “The characters are different”. Sample Input 1:
the picture was great

Sample Output 1:

The characters are same

Sample Input 2:

Hai how are you?

Sample Output 2:

The characters are different

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;



namespace program32

{

   class Program

   {

      static void Main(string[] args)

      {

         int n;

         n=UserProgramCode.checkcharacters("the picture was great");

         Console.WriteLine(n);



      }

   }

   class UserProgramCode
```

```
  {
    public static int checkcharacters(string str)
    {
      int len = str.Length;
      string str1 = str.Substring(0, 1);
      string str2 = str.Substring(len-1);
      if (str1.Equals(str2))
      {
        return 1;
      }
      else
      {
        return -1;
      }

    }
  }
}
```

125.Boundary Average

Given an int array as input, write a program to compute the average of the maximum and minimum element in the array.

Include a class UserProgramCode with a static method “getBoundaryAverage” that accepts an integer array as argument and return a avegare of max and min value.

Create a class Program which would get the input array and call the static method getBoundaryAverage present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer n, that corresponds to the size of the array.

The next n lines consist of integers that correspond to the elements in the array. Assume that the maximum number of elements in the array is 10.

Output consists of a single float value that corresponds to the average of the max and min element in the array. Output is displayed correct to 1 decimal place.

Sample Input :

6

3

6

9

4

2

5

Sample Output:

5.5

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace ConsoleApplication24
```

```
{
   class Program
   {
      static void Main(string[] args)
      {

         int n=Convert.ToInt32(Console.ReadLine());
         int[] s =new int[n];
         for(int i=0;i<n;i++)
            s[i] =Convert.ToInt32(Console.ReadLine()) ;
         double a = UserProgramCode.getBoundaryAverage(s);
         Console.WriteLine(a);
         Console.ReadLine();



      }
   }
   class UserProgramCode
   {
      public static double getBoundaryAverage(int[] a)
      {

         double d,e;
```

```
        d=((a.Max()+a.Min())/2.0);

        e = Math.Round(d, 1);


        return e;

      }

  }

}
```

126.Triplets

Given inputs ,integer array input and an integer value 'k'. Write a program to find the three elements in the array whose sum is equal to the given input value 'k' and store the output in an array. Store the output elements in the same order as present in input array. Business rule: 1) If any of the inputs is negative, then print -1. 2) If the input array does not contains the triplets whose sum is equal to integer k, then print -2. 3) If the input array contains any duplicates characters, then print -3. Create a class named UserProgramCode that has the following static method

public static int[] findTriplets(int[] input1,int k)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.

The next 'n' lines of input consist of elements in the input array. The next line of the input corresponds to the value of k.

Refer business rules and sample output for formatting specifications.

Sample Input 1 : 6 12 3 4 1 6 9 24

Sample Output 1 : 12 3 9 Sample Input 2 : 6 12 3 4 -1 6 9 16

Sample Output 2 : -1

```
using System;


class UserProgramCode
{
  public static int[] findTriplets(int[] input,int num)
  {
     int[] triplet=new int[3];
     for (int i = 0; i < input.Length; i++)
     {
        for (int j = i + 1; j < input.Length; j++)
        {
           for (int k = j+1; k < input.Length; k++)
           {
              if (input[i] < 0 || input[j] < 0 || input[k] < 0)
              {
                 triplet[0] = -1;
              }
              else if (input[i] == input[j] || input[j] == input[k] || input[i] == input[k])
              {
                 triplet[0] = -3;
              }

              else if (input[i] + input[j] + input[k] == num)
```

```
            {
                triplet[0] = input[i];
                triplet[1] = input[j];
                triplet[2] = input[k];
            }

        }
      }
    }
    if (triplet[0] + triplet[1] + triplet[2] == num)
    {
       return triplet;
    }
    else
    {
       triplet[0] = -2;
    }
    return triplet;
  }
}
using System;

class Program
```

```
{
  public static void Main( string[] args )
  {
              int size=0,i,num;
             size=Convert.ToInt32(Console.ReadLine());
     int[] arr = new int[size];
     int[] output = new int[3];
              for(i=0;i<size;i++){
              arr[i] = Convert.ToInt32(Console.ReadLine());
              }
             num = Convert.ToInt32(Console.ReadLine());
             output = UserProgramCode.findTriplets(arr,num);
             for(i=0;i<output.Length;i++)
    {
        if(output[i]!=0)
                    Console.WriteLine(output[i]);
    }
       Console.Read();

           }
  }
}
```

127. Validating the pan

Write a code to validate the given PAN number as per the following rules:

1. There must be eight characters.
2. First three letters must be alphabets followed by four digit number and ends with alphabet
3. All alphabets should be in uppercase.

Include a class UserProgramCode with static method validatePAN that accepts the String and return type should be interger.

Create a class Program which would get the input and call the static method validatePAN present in the UserProgramCode.

In validatePAN() if PAN card number is Valid return 1 otherwise return 2 . In Program

If the method returns 1 then print "Valid PAN code" If the method returns 2 then print "Invalid PAN code" Sample Input 1 ALD3245E Sample Output 1 Valid PAN code Sample Input 2 4567123 Sample Output 2

Invalid PAN code

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;


namespace Validate_pan

{

   class UserPRogramCode

   {

      public static int ValidatePan(string input1)

      {
```

```
        int output1 = 0;
        Regex ex = new Regex("^([A-Z]{3}[0-9]{4}[A-Z]{1})$");
        if (ex.IsMatch(input1))
        {
           output1 = 1;
        }
        else
        {
           output1 = 2;
        }
        return output1;
      }
   }
}
  class Program
   {
      static void Main(string[] args)
      {
         string str = Console.ReadLine();
         int op = UserPRogramCode.ValidatePan(str);
         if (op == 1)
            Console.WriteLine("Valid PanCode");
         else
```

```
            Console.WriteLine("Invalid");
            Console.ReadLine();
        }
    }
}
```

128. Count Characters

Write a program to count the number of characters present in the given input String array. Include a class UserProgramCode with static method countCharacters which accepts string array. The return type is a integer value which is the count of characters in the string array. Create a class Program which would get the input and call the static method countCharacters present in the UserProgramCode. Input string must contains only the alphabets then return count of characters else return the -1. If count value is -1 then print "Invalid Input". Input and Output Format : Input consists of a integer and String array. Integer represents a size of the array following by the string elements. Output consists of a integer which is the count of the character from string array. Sample Input 1: 3 cherry apple blueberry Sample Output 1: 20 Sample Input 2: 2 @aaa bbb Sample Output 2: Invalid Input

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication18
{
    class UserProgramCode
    {
```

```
public static int countcharachters(string[] s)
{
   int  sum = 0,flag = 0 ;

   foreach (string s1 in s)
   {
      char[] ch = s1.ToCharArray();
      foreach (char c in ch)
      {
         if (char.IsLetter(c))
         {
            flag++;
         }

      }
   }

   foreach (string s1 in s)
   {
      sum = sum + s1.Length;
   }
   if(flag==sum)
      return sum;
```

```
        else
            return -1;

    }
    }
}


  class Program
    {
        static void Main(string[] args)
        {
            UserProgramCode u=new UserProgramCode();
            int n = int.Parse(Console.ReadLine());
            string[] s=new string[n];
            int result;
            for (int i = 0; i < n; i++)
                s[i] = Console.ReadLine();
            result=UserProgramCode.countcharachters(s);
            if(result==-1)
                Console.WriteLine("Invalid Input");
            else
```

```
        Console.WriteLine(result);
    }


  }
}
}
```

129. String Occurences

Write a program to count the number of occurences of second word of second sentence in the first sentence.

Return the count as output. Note - Consider case. Include a class UserProgramCode with a static method countNoOfWords which accepts two string variables. The return type is the integer.

Create a Class Program which would be used to accept two Input strings and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of two strings with maximum size of 100 characters.

Output consists of an integer.

Refer sample output for formatting specifications.

Sample Input 1:

abc bcd abc bcd abc abc

av abc

Sample Output 1:

4

Sample Input 2:

ABC xyz AAA

w abc

Sample Output 2:

0

```
using System;
```

```
class UserProgramCode
{
  public static int countNoOfWords(string str1,string str2)
  {
            string[] sub1=str1.Split(' ');
    string[] sub2 = str2.Split(' ');
      int ctr=0;



    string str = sub2[1];


    for (int i = 0; i < sub1.Length; i++)
    {
       if (sub1[i] == str)
       {
          ctr++;
       }


    }
    return ctr;



  }
```

```
}
class Program
{
  public static void Main( string[] args )
  {
        string str1=Console.ReadLine();
        string str2=Console.ReadLine();
        Console.WriteLine(UserProgramCode.countNoOfWords(str1,str2));
    Console.ReadLine();
  }
}
```

130. Find Common Elements

Write a program to find the common elements from the 2 integer lists and to print them in ascending order.

Example:
input1: [4,7,3,9,1,5]
input2: [10,4,6,5,3]

Output1:[3,4,5]

Business Rules:

Only positive numbers should be given to the input Lists.

1. If the input1 List consists of negative numbers, store -1 in the list returned from the method..
2. If the input2 List consists of negative numbers, store -2 in the list returned from the method.
3. If both the input lists, input1 and input2 consists of negative numbers,store -3 in the list returned

from the method.

Include a class UserProgramCode with a static method FindCommonElements which accepts two integers (size of the two integer list), and the two integer lists. The return type (integer list) should return the output integer list. Store -1,-2 or -3 in the output list according to the business rules and return if necessary.

Create a Class Program which would be used to accept two integers (size of the two integer list), and the two integer lists, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of two integers (size of the two integer list), and the two integer lists.

Output consists of an Integer list or, a String “First list is negative” if -1 is returned, “ Second list is negative” if -2 is returned, or “Both lists are negative” if -3 is returned.

Refer sample output for formatting specifications.

Sample Input 1:

6

5

4

7

3

9

1

5

10

4

6

5

3

Sample Output1:

3

4

5

Sample Input 2:

6

5

-4

7

3

9

1

5

10

4

6

5

3

Sample Output2:

First list is negative

Sample Input 3:

6

5

4

7

3

9

1

5

10

4

6

5

-3

Sample Output 3:

Second list is negative

Sample Input 4:

6

5

-4

7

3

9

1

5

10

4

6

5

-3

Sample Output 4:

Both lists are negative

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace fFindCommon
```

```
{
    class UserProgramCode
    {
        public static List<int> FindCommon(List<int> list1, List<int> list2)
        {
            List<int> final=new List<int>();
            int flag = 0;
            int flag1 = 0;
            for (int i = 0; i < list1.Count; i++)
            {
                if (list1[i] < 0)
                {
                    final.Add(-1);
                    flag = 1;


                }
            }
            for (int i = 0; i < list2.Count; i++)
            {
                if (list2[i] < 0)
                {
                    final.Add(-2);
                    flag1 = 2;
```

```
    }
}
if (flag == 1 && flag1 == 2)
{
    final.Clear();
    final.Add(-3);
    return final;
}
else if (flag == 1)
    return final;
else if(flag1==2)
    return final;
else
for (int i = 0; i < list1.Count; i++)
{
    for (int j = 0; j < list2.Count; j++)
    {
        if(list1[i]==list2[j])
        {
            final.Add(list1[i]);
        }
    }
}
```

```
            final.Sort();
            return final;

        }
    }
}
    class Program
    {
        static void Main(string[] args)
        {
            int n1 = int.Parse(Console.ReadLine());
            List<int> list = new List<int>();
            for (int i = 0; i < n1; i++)
            {
                list.Add(Convert.ToInt32(Console.ReadLine()));

            }
            int n2 = int.Parse(Console.ReadLine());
            List<int> list2 = new List<int>();
            for (int i = 0; i < n2; i++)
            {
                list2.Add(Convert.ToInt32(Console.ReadLine()));
```

```
        }


        List<int> op = UserProgramCode.FindCommon(list, list2);
        foreach (int item in op)
        {
           Console.WriteLine(item);
        }
        Console.ReadLine();
      }
   }
}
```

131. Longest Word

Write a Program which finds the longest word from a sentence. Your program should read a sentence as input from user and return the longest word. In case there are two words of maximum length return the word which comes first in the sentence.

Include a class UserProgramCode with a static method getLargestWord which accepts a string. The return type is a string that corresponds to the largest word in the sentence.

Create a Class Program which would be used to accept a input string and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a string with a maximum size of 100 characters.

Output consists of a single string.

Refer sample output for formatting specifications.

Sample Input 1:

Welcome to the world of Programming

Sample Output 1:

Programming

Sample Input 2:

ABC DEF

Sample Output 2:

ABC

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace level1_57
{
   class UserProgramCode
   {
      public static string getLargestWord(string str)
      {
         int l=0,max=0,ind=-1;
         string[] s=new string[100];
         s = str.Split(' ');
         for (int i = 0; i < s.Length; i++)
         {
            l = s[i].Length;
            if (max < l)
            {
```

```
                max = l;
                ind = i;
              }
          }
          return s[ind];

      }
   }
   class Program
   {
      static void Main(string[] args)
      {
         string str,str1;
         str = Console.ReadLine();
         str1 = UserProgramCode.getLargestWord(str);
         Console.WriteLine(str1);
         Console.Read();
      }
   }
}
```

132. Largest Digit

Write a program to find the Largest digit from given input integer. Print the largest digit. If the number is negative, print “Negative Number”.

Example:

Input1: 524

Output1: 5

Include a class UserProgramCode with a static method findLargestDigit which accepts an integer. The return type (integer) should return the largest digit. Return -1 if the number is negative.

Create a Class Program which would be used to accept an integer and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of an integer.

Output consists of an integer (the largest digit), or a String " Negative Number" if the input is negative.

Refer sample output for formatting specifications.

Sample Input 1:

524

Sample Output 1:

5

Sample Input 2:

-23

Sample Output 2:

Negative Number

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication100
```

```
{
    class Program
    {
        static void Main(string[] args)
        {
            int a;
            a = int.Parse(Console.ReadLine());
            int c = UserProgramCode.findLargestDigit(a);
            if(c==-1)
                Console.WriteLine("Negative Number");
            else
                Console.WriteLine(c);
        }
    }
    class UserProgramCode
    {
        public static int findLargestDigit(int b)
        {
            if (b >= 0)
            {
                string a = b.ToString();
                int l = a.Length;
                int c = 0;
```

```
            for (int i = 0; i < l; i++)
            {
               int x = b / 10;
               int y = b % 10;
               b = x;
               if (y >= c)
               {
                  c = y;
               }
            }
            return c;
         }
         else
            return (-1);
      }
   }
}
```

133. Dash Check

Write a program to read two strings and check whether or not they have dashes in the same places. Print “Yes” if the condition satisfies, else print “No”.

Include a class UserProgramCode with a static method compareDashes which accepts two strings and returns an integer. The function returns 1 if all dashes are placed correctly, else the function returns 2.

Create a Class Program which would be used to accept two strings and call the static method present in UserProgramCode.

Note: The strings must have exactly the same number of dashes in exactly the same positions. The strings might be of different length.

Input and Output Format:

Input consists of two strings.

Output consists of a string (“Yes” or “No”).

Refer sample output for formatting specifications.

Sample Input 1:

hi—there-you.

12--(134)-7539

Sample Output 1:

Yes

Sample Input 2:

-15-389

-xyw-zzy

Sample Output 2:

No

```
class Program
  {
     public static void Main(string[] args)
     {
        string input1, input2;
        input1 = Console.ReadLine();
        input2 = Console.ReadLine();
        int value = UserProgramCode.compareDashes(input1, input2);
        if (value == 1)
```

```
            Console.WriteLine("Yes");
        else
            Console.WriteLine("No");


        Console.ReadLine();
    }
  }
class UserProgramCode
  {
    public static int compareDashes(string input1, string input2)
    {
      // fill your code here



      char[] s1 = input1.ToCharArray();
      char[] st1 = input2.ToCharArray();
      int len, count = 0;
      if (s1.Length > st1.Length)
      {
        len = s1.Length;
      }
      else
      {
```

```
    len = st1.Length;
}
for (int i = 0; i < len; i++)
{
    if (s1[i] == '-')
    {
        if (st1[i] != '-')
        {
            count++;
            break;
        }
    }
}
if (count > 0)
    return 2;
else
    return 1;
```

134. Arrange After Cubing

Write a code to insert cube of a number in between two numbers in an integer array if those numbers satisfy the below conditions: Conditions: 1) The elements in the array must be consecutive numbers 2) Second element in the array should be the square of first element in the array. 3) Cube of first number must be inserted in between the first element and second element. 4) If any of the element in the array does not satisy the above condition add the element in the output array and proceed with the next element. Business Rules: 1. If no consecutive numbers are present assign all the elements in the Input array to the Output array . 2. If input array consists of negative elements perform the square of negative

numbers and then print the final output array. Create a class named UserProgramCode that has the following static method

public static int[] arrangeAfterCubing(int[] input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.

The next 'n' lines of input consist of elements in the input array.

Refer sample output for formatting specifications.

Sample Input 1: 7 1 2 4 6 7 3 9 Sample Output 1: 1 2 8 4 6 7 3 27 9 Sample Input 2:

3 1 -4 5

Sample Output 2: 1 16 5

```
using System;


class Program
  {
     public static void Main( string[] args )
     {
                        int n=Convert.ToInt32(Console.ReadLine());
                        int[] value=new int[n];
                        for(int i=0;i<n;i++)
                        {
                                value[i]=Convert.ToInt32(Console.ReadLine());

        }
```

```
                int[] output = UserProgramCode.arrangeAfterCubing(value);
                for(int i=0;i<output.Length;i++)
                {
                        Console.Write(output[i]+"\n");
                }
        }
  }
```

```
using System;
using System.Text;
using System.Collections;
class UserProgramCode
{
  public static int[] arrangeAfterCubing(int[] input1)
  {
     int i=0,ch,c;
     int ch1=0,flag=1;
     int s = input1.Length;
     ArrayList sb = new ArrayList();
```

```
int[] op1 = new int[s];
for (i = 0; i < s; i++)
{
   if (input1[i] < 0)
   {
      flag = 0;
      ch1 = input1[i] * input1[i];
      input1[i] = ch1;
   }

}
if (flag == 0)
{
   for (i = 0; i < s; i++)
   {
      op1[i] = input1[i]; ;

   }
   return op1;

}
else
{
```

```
for (i = 0; i < s - 1; i++)
{
   ch = input1[i] * input1[i];
   if (input1[i + 1] == ch)
   {
      c = input1[i] * input1[i] * input1[i];
      if (i == 0)
      {
         sb.Add(input1[i]);
         sb.Add(c);
         sb.Add(input1[i + 1]);
      }
      else
      {
         sb.Add(c);
         sb.Add(input1[i + 1]);
      }
   }
   else
   {
      if (i == 0)
      {
         sb.Add(input1[i]);
```

```
                sb.Add(input1[i + 1]);
            }
            else
            {
                sb.Add(input1[i + 1]);
            }
        }
    }
    int len = sb.Count;
    int[] op = new int[len];
    i = 0;
    foreach (int n in sb)
    {
        op[i] = n;
        i++;
    }
    return op;
  }
 }
}
```

135. String Equal Check

Given two strings Input1 and Input2 and integer Input3, write a program to check if Nth character of Input1 traversing from first and Nth character of Input2 traversing from last are same irrespective of

case where N is the Input3 value. Ignore case. If both are same, then print "The character is x" where x is the Nth character If both are not same, then print "The character x and y does not match" where x is the Nth character of Input1 starting from first and y is the Nth character of Input2 starting from last. Business rule: 1) If the Input1 string contains any special characters or numbers, then print 'Invalid Input' 2) If the Input2 string contains any special characters or numbers, then print 'Invalid Input' 3) If the Input3 value is greater than the length of Input1 and/or Input2, then print 'Invalid Input' Create a class named UserProgramCode that has the following static method

public static string stringEqualCheck(string input1, string input2, int input3)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:

The first line of the input consists of a string that corresponds to input1.

The second line of the input consists of a string that corresponds to input2.

The third line of the input consists of an integer that corresponds to input 3.

Output consists of a string. Refer business rules and sample output for the format.

Sample Input 1: Battle Final 2 Sample Output 1:

The character is a Sample Input 2 : Photograph Anticipate 4

Sample Output 2:

The character t and p does not match Sample Input 3 : xerox pretty 15 Sample Output 3 :

Invalid Input

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

namespace ConsoleApplication23

{
```

```
class UserProgramCode
{
   public static string stringEqualCheck(string input1, string input2, int input3)
   {
      string i1, i2;
      int i3;

      i1 = input1;
      char[] i11 = i1.ToCharArray();
      i2 = input2;
      char[] i22 = i2.ToCharArray();
      i3 = input3;
      foreach (char ch in i11)
      {
         if (char.IsDigit(ch))
         {
            Console.WriteLine("invalid input");
         }
         if (!char.IsLetter(ch))
         {
            Console.WriteLine("invalid input");
         }
      }
```

```
foreach (char ch in i22)
{
   if (char.IsDigit(ch))
   {
      return "invalid input";
   }
   if (!char.IsLetter(ch))
   {
      return "invalid input";
   }
}
if (i3 > i1.Length)
{
   return "invalid input";
}
else if (i3 > i2.Length)
{
   return "invalid input";
}
```

```
        else if ((i11[i3 - 1]) == (i22[(i2.Length) - i3]))
        {
           return "the character is " + i11[i3 - 1];
        }
        else
        {
           return "the character " + i11[i3 - 1] + " and " + i22[(i2.Length) - i3] + " does not match";
        }




      }
   }
}
############################
      using System;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using TestPractice;
using ConsoleApplication23;
//using ConsoleApplication9;
```

```
namespace TestPractice
{
    class Program
    {
        static void Main(string[] args)
        {
            string s1, s2;
            int index;
            s1 = Console.ReadLine();
            s2 = Console.ReadLine();
            index = int.Parse(Console.ReadLine());
            string disp;
            disp = UserProgramCode.stringEqualCheck(s1, s2, index);
            //if(disp=="1")
            //    Console.WriteLine("the character is " + i11[i3 - 1]);


            Console.WriteLine(disp);
            Console.ReadLine();


        }
    }
}
```

136. Calculate VAT

ABC stores needs a computerized solution for calculating the VAT for the billing amount. Write a code to calculate the VAT amount for the input bill amount. The VAT should be calculated in the following basis, Type ------- VATPercentage, Medical (M) - - - - - - - - - -- 9% Vegetables (V) and fruits - - - - - - - - - - 5% Clothes (C) - - - - - - - - - - - - 12% Electronics (E) - - - - - - - - - - -- - - 6.25% Business Rules : 1. The codes 'M','V','C' or 'E' only should be given as input for indicating the Medical, Vegetables and Fruits, Clothes and Electronics type of goods respectively. Any other character other than the above is given as input, it is Invalid Input. 2. Only Positive number should be given as a input for bill amount. Else it is Invalid Input. Include a class UserProgramCode with a static method calculateVAT which accepts a character and double and returns a double which corresponds to the calculated VAT amount. If the input is invalid, the method returns -1.

Create a class Program which would get the input and call the static method calculateVAT present in the UserProgramCode. If the method returns -1, print 'Invalid Input'. Input and Output Format: Input consists of character and double. Character denotes a goods type and double denotes total amount. Refer sample output for formatting specifications. Sample Input 1 : M 70 Sample Output 1 : 6.3 Sample Input 2 : V -500 Sample Output 2 : Invalid Input

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace final

{

   class UserProgramCode

   {

      public static double CalculateVat(char c, double input1)
```

```
{
   double percent=0;
   double vat=0;
   double output1=0;
   if (input1 < 0)
      return -1;
    switch (c)

   {

      case 'M':

          percent = 9;

          vat = percent / 100;



          output1 = input1* ( vat);

         return output1;
```

```
case 'V':

    percent = 5;

    vat = percent / 100;

    output1 = input1 * ( vat);


   return output1;


case 'C':

    percent = 12;

    vat = percent / 100;

    output1 = input1 * ( vat);

   return output1;
```

```
    case 'E':

        percent = 6.25;

        vat = percent / 100;

        output1 = input1 * ( vat);

       return output1;



    default:

        output1 = -1;


        break;

}

 return output1;
```

```
        }


    }

}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace final
{
    class Program

    {
        static void Main(string[] args)
        {
            Char c = Convert.ToChar(Console.ReadLine());
            double input1 = Convert.ToDouble(Console.ReadLine());
```

```
        double op = UserProgramCode.CalculateVat(c, input1);
        if (op == -1)
           Console.WriteLine("Invalid Input");
        else
        Console.WriteLine(op);
        Console.ReadLine();



      }
   }
}
```

137. Get word with Maximum Vowels

Write a method that accepts a string input and returns the word with maximum number of vowels. In case there are two or more words with maximum number of vowels, return the first word.

Example:
Input: Appreciation is the best way to motivate
Output: Appreciation (total vowels = 6)

Include a class UserProgramCode with a static method getWordWithMaximumVowels that accepts a string and returns a string.

Create a class Program which would get the input and call the static method getWordWithMaximumVowels present in the UserProgramCode.

Input and Output Format:
Input consists of a string.
Output consists of a string. Sample Input1 : Appreciation is the best way to motivate Sample Output1 :

Appreciation Sample Input2 : Sun rises in the east Sample Output2 : rises

```
using System.Linq;
using System.Text;

namespace get_the_word_with_max_vowles
{
   class Program
   {
      static void Main(string[] args)
      {
         string str = Console.ReadLine();
         string op = UserProgramcode.GETMAXVOWELS(str);
         Console.WriteLine(op);
         Console.ReadLine();



      }
   }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace get_the_word_with_max_vowles
{
   class UserProgramcode
   {
      public static string GETMAXVOWELS(string str)
      {
         string[] arr = str.Split(' ');
         int max = 0, lenmax = 0, count;
         foreach (string element in arr)
         {

            char[] ch = element.ToCharArray();
            count = 0;
            for (int i = 0; i < ch.Length; i++)
            {
               if (ch[i] == 'a' || ch[i] == 'e' || ch[i] == 'i' || ch[i] == 'o' || ch[i] == 'u')
                  count++;
```

```
        }
        if (count > max)
        {
            max = count;

            lenmax = ch.Length;
        }

    }

    string output = string.Empty;
    foreach (string item in arr)
    {
        if (item.Length == lenmax)
        {
            output = item;
            break;
        }
    }
    return output;
}
```

```
    }
}
```

138. Sort the list

Write a program which reads an Integer(size of the list), a String List and a character, and to get the strings that will not start with the given character irrespective of case. Sort the elements in ascending order based on its length. Print the output list. If the elements are having the same length, then display the elements in alphabetical order.

Include a class UserProgramCode with static method GetTheElements which accepts a String list and a character. The return type is List<String>.

Create a Class Program which would be used to get the inputs and call the static method present in UserProgramCode. In GetTheElements method

Only alphabets should be given in list , otherwise return "Invalid Input". When the output list is empty, then return "List is Empty". Otherwise return the appropriate result.

In Program class Print the result which is return by GetTheElements method in UserProgramCode. Input output format The first line of the input is an integer that corresponds to n, the size of the list. The next n lines of input correspond to the elements in the string list. The line of the input contains the character. The output is the List type List<String>. Sample Input 1: 3 read write edit e Sample Output 1: read write Sample Input2 : 2 Elegent event e Sample Output2 : List is Empty Sample Input 3: 2 Eleg$ent e^ent e Sample Output 3 : Invalid Input

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace progm55
{
    class UserProgramCode
    {
```

```
public static int[] sortList(int[] a)
{
   int temp;

   int maxi = a.Length;

   if (maxi == 0)
  {
      a[0] = -1;
   }
   else
   {

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


            if (a[i] < 0)
            {
               a[0] = -1;
```

```
            }


            else
            {
               if (a[i] < a[j])
               {
                  temp = a[i];
                  a[i] = a[j];
                  a[j] = temp;


               }



            }
         }
      }


   }


      return a;


}
```

```
 }
class Program
 {
    static void Main(string[] args)
    {

        int n = Convert.ToInt32(Console.ReadLine());
        int[] a=new int[n];
        int[] output=new int[n];
       for (int i = 0; i < n; i++)
        {
            a[i] = Convert.ToInt32(Console.ReadLine());

        }
        output=UserProgramCode.sortList(a);



            if (output[0] != -1)
            {
                for (int i = 0; i < n; i++)
                {
                Console.WriteLine(output[i]);
```

```
            }
        }
        else
        {
            Console.WriteLine("Invalid Input");
        }

    }


}
}
```

139. Matching String

Write a program to display the strings which starts with the character passed in input2 variable from input1 list irrespective of case. Store the result in output list in the same sequence in which they are found in input1 list. Then form a string in output1 list as in the example given below. In the example given below, 2 corresponds to the numer of strings in the input list that start with the given character.

Example:

input1: [abc,apple,Mango]
input2: a

Output1:[abc_2,apple_2]

Business Rule:

1. If there is no match found in input1 list then store "-1" in the string list returned from from the method and print “No match found” in Main.

Include a class UserProgramCode with a static method sortStrings which accepts an integer (the size of the string list), a String list and a character. The return type (String List) should return the output String List.

Create a Class Program which would be used to accept an integer, a String List and a character , and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of an integer, a String list and a character, where the integer corresponds to the size of the List, String list corresponds to the input string list and the character values corresponds to the starting character.

Output consists of a String List.

Refer sample output for formatting specifications.

Sample Input 1:

3

abc

apple

Mango

a

Sample Output 1:

abc_2

apple_2

Sample Input 2:

3

abc

apple

Mango

b

Sample Output 2:

No match found

```
using System;
using System.Collections.Generic;


class Program
{
   public static void Main(string[] args)
   {
      List<string> list = new List<string>();
      List<string> result = new List<string>();
      int size = Convert.ToInt32(Console.ReadLine());
      for (int i = 0; i < size; i++)
      {
         list.Add(Console.ReadLine());
      }
      char ch = Convert.ToChar(Console.ReadLine());
      result = UserProgramCode.SortStrings(size, list, ch);
      if (result[0].Equals("-1"))
         Console.WriteLine("No match found");
      else
      {
```

```
        for (int i = 0; i < result.Count; i++)
            Console.WriteLine(result[i]);
        }

    }
}

using System;
using System.Collections.Generic;
class UserProgramCode
{
    public static List<string> SortStrings(int size, List<string> li, char ch)
    {
        List<string> result = new List<string>();
        // int size = Convert.ToInt32(Console.ReadLine());
        int k = 0, count = 0;

        for (int i = 0; i < size; i++)
        {
            char c = Convert.ToChar(li[i].Substring(0, 1));
            if (ch == c)
                count++;
        }
```

```
        for (int i = 0; i < size; i++)
        {
            string item;
            char c = Convert.ToChar(li[i].Substring(0, 1));
            if (ch == c)
            {
                item = li[i] + "_" + count;
                result.Add(item);
                k++;
            }

        }
        return result;
    }
}
```

Matching string(SortStrings)

```
using System;
```

```
using System.Collections.Generic;


class Program
{
   public static void Main(string[] args)
   {
      List<string> list = new List<string>();
      List<string> result = new List<string>();
      int size = Convert.ToInt32(Console.ReadLine());
      for (int i = 0; i < size; i++)
      {
         list.Add(Console.ReadLine());
      }
      char ch = Convert.ToChar(Console.ReadLine());
      result = UserProgramCode.SortStrings(size, list, ch);
      if (result[0].Equals("-1"))
         Console.WriteLine("No match found");
      else
      {
         for (int i = 0; i < result.Count; i++)
            Console.WriteLine(result[i]);
      }
   }
```

```
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
class UserProgramCode
{
    public static List<string> SortStrings(int size, List<string> li, char ch)
    {
        List<string> res = new List<string>();

        int c = 0;

        foreach (string item in li)
        {
            if (item.Contains(ch))
            {
                c++;
            }
        }
        c = c - 1;
        li.RemoveAt(size - 1);
```

```
        if (c == 0 )
        {
            res.Add("-1");
            return res;
        }

        else
        {
            foreach (var item in li)
            {
                if (item.Contains(ch))
                {
                    res.Add(item + "_" + c);
                }
            }
        }

        return res;

    }

}
```

140. Count Sequential Characters

Get a string as input and write code to count the number of characters which gets repeated 3 times consecutively and return that count (ignore case).

Include a class UserProgramCode with a static method countSequentialChars which accepts a string as input and return type is an integer.

The method returns the repeat count. If no character gets repeated 3 times consecutively the method returns -1.

Create a class Program which would get the input and call the static method countSequentialChars present in the UserProgramCode. If the method returns -1, print 'No Repeated Words Found'.

Input and Output Format:

Input consists a string.

Refer sample output for formatting specifications.

Sample Input 1:

abcXXXabc

Sample Output 1:

1

Sample Input 2:

aaxxyzAAx

Sample Output 2:

No Repeated Words Found

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;
```

```
namespace ConsoleApplication24
{
   class Program
   {
      static void Main(string[] args)
      {
         string s = Console.ReadLine();
         int a = UserProgramCode.countSequentialChars(s);
         if(a==-1)
            Console.WriteLine("No Repeated Words Found");
         else
            Console.WriteLine(a);
         Console.ReadLine();

      }
   }
   class UserProgramCode
   {
      public static int countSequentialChars(string s)
      {

         int l = s.Length;
         string[] st = new string[50];
```

```
for (int k = 0; k < l; k++)
{
   st[k] = s.Substring(k, 1);



}

int count = 0;
int c=0;

for (int k = 0; k < l - 1; k++)
{

      if (st[k] == st[k+1])
         count++;
      else
         count = 0;
      if (count == 2)
         c++;


   }
```

```
        if(c==0)
        return -1;
        else
           return c;
      }
   }
}
```

141. Check Supply

The Policy followed by a Company to process Customer Orders is given by the following rules:

Rules:

(a) If a Customer Order is less than or equal to that in Stock and if the Credit is OK, Supply the required quantity.

(b) If the Credit is Not OK, then do not Supply. Send him intimation saying "Cannot Supply".

(c) If the Credit is OK, but the item in Stock is less than the order, Supply what is in Stock. Intimate to him that the balance will be shipped.

(c) If the Credit is OK and the item in Stock is 0, Output should be "Out Of Stock".

Input1- Stock in hand

Input2- Customer Order Quantity

Input3- Credit (true -OK, false -Not OK)

Output1- Message(“Supply","Cannot Supply","Out Of Stock","Balance Will Be Shipped Later")

Example:

Input1: 50

Input2: 5

Input3: true

Output1: Supply

Include a class UserProgramCode with a static method checkSupply which accepts two Integers and a Boolean value. The return type (String) should return a String according to the business rules.

Create a Class Program which would be used to accept two Integers and a Boolean value, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a 2 integers and a boolean value.

Output consists of a String.

Refer sample output for formatting specifications.

Sample Input 1:

50

5

true

Sample Output 1:

Supply

Sample Input 2:

50

5

false

Sample Output 2:

Cannot Supply

Sample Input 3:

50

55

true

Sample Output 3:

Balance Will Be Shipped Later

Sample Input 4:

5

true

Sample Output 4:

Out Of Stock

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace ConsoleApplication18
{
   class UserProgramCode
   {
      public static string checksupply(int n1, int n2,bool value)
```

```
{
   if (value == true && n1 == 0)
   {
      return ("OutOfStock");
   }
   else

      if (value == true && n2 < n1 | n1 == n2)
      {
         return ("Supply");
      }
      else

         if (value == true && n1 < n2)
         {
            return ("Balance Will Be Shipped Later");
         }

         else
            if (value == false)
            {
               return "Cannot Supply";
            }
```

```
                else
                    return "";



        }



    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication18
{
  class Program
   {
       static void Main(string[] args)
       {
           UserProgramCode u = new UserProgramCode();
           int n1, n2;
```

```
        bool value;
        string output;
        n1 = int.Parse(Console.ReadLine());
        n2=int.Parse(Console.ReadLine());
        value=bool.Parse(Console.ReadLine());
        output = UserProgramCode.checksupply(n1, n2, value);
        Console.WriteLine(output);

    }

  }
}
```

142. Reverse Substring

Given a input string with a startIndex and length, Write a program to extract substring from right to left. Assume the last character has index 0.

Include a class UserProgramCode with a static method reverseSubstring which accepts a string and two integers. The return type is string as given in the above statement.

Create a Class Program which would be used to accept Input and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a string, and two integers – startIndex and length.

Output consists of a string as mentioned in the problem statement.

Refer sample output for formatting specifications.

Sample Input 1:

rajasthan

2

3

Sample Output 1:

hts

```
using System;


        using System.Collections.Generic;

        using System.Linq;

        using System.Text;


namespace ConsoleApplication9

{

   public class UserProgramCode

   {

      public static string reverseSubstring(string str, int start, int len)

      {

         //StringBuilder sb = new StringBuilder();

         char[] ch = str.ToCharArray();

         Array.Reverse(ch);

         string s=new string(ch);

          string outp = s.Substring(start, len);
```

```
            //foreach (char item in ch)
            //{
            //    sb.Append(item);
            //}

             return outp;



        }
    }
}
                using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using TestPractice;
using ConsoleApplication9;

namespace TestPractice
{
class Program
                {
                    static void Main(string[] args)
```

```
        {
    string str=Console.ReadLine();
    int start = int.Parse(Console.ReadLine());
    int len = int.Parse(Console.ReadLine());
    string str2 = UserProgramCode.reverseSubstring(str, start, len);
    Console.WriteLine(str2);
    Console.ReadLine();
        }
      }
   }
```

143. Add Years

Write a method which can display the date n years after the given date. The date should be given in string format "mm/dd/yyyy" without time and the resultant added date should also be in the format "mm/dd/yyyy" . “n” as Years and the given date should be accepted as an argument. Include a class UserProgramCode with a static method addYears which accepts a string as input and output should be a string.

Business Rules : 1) Only positive value should be given as input to the integer else the method returns -2. 2) If the date format is not "mm/dd/yyyy" given in the string, the method returns -1. 3) Otherwise return the corresponding date in given format. Create a class Program which would get the input and call the static method addYears present in the UserProgramCode. If the method returns -1, print 'Invalid date format'. If the method returns -2, print 'Invalid Input'. Input and Output Format : Input consists of a string which accepts a date. Refer the sample input and output for formatting specifications. Sample Input 1 : 10/21/2009 5 Sample Output 1 : 10/21/2014 Sample Input 2 : 2009/10/03 -2 Sample Output 2 : Invalid Input Sample Input 3 : 27-10-2009 1 Sample Output 3 : Invalid date format

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Globalization;
namespace AddYears
{
   class Program
   {
      public static void Main(string[] args)
      {
         string inputDate = Console.ReadLine();
         int day = Convert.ToInt32(Console.ReadLine());
         string output = UserProgramCode.addYears(inputDate, day);
         if (output.Equals("-1"))
            Console.WriteLine("Invalid Input");
         else if (output.Equals("-2"))
            Console.WriteLine("Invalid date format");
         else
            Console.WriteLine(output);
         Console.ReadLine();
      }
   }
```

```
class UserProgramCode
{
    public static string addYears(string date, int day)
    {
        // fill your code here
        DateTime dt;
        string output;

        if (day < 0)
        {
            return "-1";
        }
        else
        {
            bool res = DateTime.TryParseExact(date, "MM/dd/yyyy", null,
System.Globalization.DateTimeStyles.None, out dt);

            if (res == true)
            {
                dt = dt.AddDays(day);

                output = dt.ToString("MM/dd/yyyy");
                return output;
```

```
            }
            else
            {
               return "-2";

            }
         }

      }
   }
}
```

144. Count Characters

Given a string array as input, write a program to find the total number of characters in all the words (in the given string array). Assume that all strings are single words.

Create a class named UserProgramCode that has the following static method

public static int countCharacters(string[] input1)

Create a class named Program that accepts a string array as input and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer 'n' that corresponds to the number of elements in the string array.

The next 'n' lines of the input consists of strings that correspond to the elements in the string array.

Output consists of a single integer that corresponds to the total number of characers in all the words in the given string array.

Sample Input : 2 cherry apple Sample Output : 11

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication18
{
   class UserProgramCode
   {
      public static int countcharachters(string[] s)
      {
         int  sum = 0,flag = 0 ;

         foreach (string s1 in s)
         {
            char[] ch = s1.ToCharArray();
            foreach (char c in ch)
            {
               if (char.IsLetter(c))
               {
                  flag++;
               }

            }
```

```
        }

        foreach (string s1 in s)
        {
           sum = sum + s1.Length;
        }
        if(flag==sum)
           return sum;

        else
           return -1;

    }
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication18
{
```

```
class Program
{
    static void Main(string[] args)
    {
        UserProgramCode u=new UserProgramCode();
        int n = int.Parse(Console.ReadLine());
        string[] s=new string[n];
        int result;
        for (int i = 0; i < n; i++)
            s[i] = Console.ReadLine();
        result=UserProgramCode.countcharachters(s);
        if(result==-1)
            Console.WriteLine("Invalid Input");
        else
        Console.WriteLine(result);
    }

}
}
```

145. Find Gift Voucher

In a game two dice is thrown. From the sum of the two dice, the player is going to get the gift voucher from the club. Write a program to find the amount of the gift voucher. Print the amount received as gift.

Sum of Two Dices ---------- Gift Voucher in Rs

2,3,6,11 ---------- 1000

4,7,10---------- 3000

5,8,9,12---------- 5000

In the method,

Only Positive number (1-6) should be given as a input numbers. Else return -1.

Include a class UserProgramCode with a static method findGiftVoucher which accepts two integers. The return type (Integer) should return the gift voucher amount. If the any of the inputs is invalid return -1.

Create a Class Program which would be used to accept a positive Integer, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of two integers.

Output consists of an Integer(the gift voucher amount) or a String “Invalid Input” if any of the inputs is invalid.

Refer sample output for formatting specifications.

Sample Input 1:

1

2

Sample Output 1:

1000

Sample Input 2:

1

-2

Sample Output 2:

Invalid Input

```
using System;
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using TestPractice;


namespace TestPractice
{
   class Program
   {
      static void Main(string[] args)
      {
         int n, c, m;
         n = Convert.ToInt32(Console.ReadLine());
         m = Convert.ToInt32(Console.ReadLine());
         c = userprogramcode.findGiftVoucher(n, m);
         if (c == -1)
         {
            Console.WriteLine("invalid input");
         }
         else
         {
            Console.WriteLine(c);
```

```
        }
        Console.ReadLine();

    }
  }
}
```

###############

```
  class userprogramcode

  {
    public static int findGiftVoucher(int a, int b)
    {
      if (a > 0 && b > 0 && a < 7 && b < 7)
      {
        if ((a + b == 2) || (a + b == 3) || (a + b == 6) || (a + b == 11))
          return (1000);
        else if ((a + b == 4) || (a + b == 7) || (a + b == 10))
          return (3000);
        else if ((a + b == 5) || (a + b == 8) || (a + b == 9) || (a + b == 12))
          return (5000);
```

```
        }
        else return (-1);
        return 0;
      }

  }

}
```

146. Maximum Vowels

Given a sentence as string input, write a program to fetch the word with maximum number of vowels and print it. In case there are two or more words with maximum number of vowels, print the first word.

Create a class named UserProgramCode that has the following static method

public static string getWordWithMaximumVowels(string input1) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of a string.

Refer business rules and sample output for formatting specifications. Sample Input : Appreciation is the best way to motivate Sample Output :

Appreciation

```
using System;


class Program
{
  public static void Main( string[] args )
  {
```

```
    string input = Console.ReadLine();
    string result = UserProgramCode.getWordWithMaximumVowels(input);
    Console.WriteLine(result);
    Console.ReadLine();
  }
}
```

```
using System;

class UserProgramCode
{
   public static string getWordWithMaximumVowels(string inpt)
  {
    string result = "";

    int maxvowels = 0;
      string input=inpt.ToLower();

    string[] array = input.Split();
    foreach (string str in array)
```

```
    {
        int vowels = 0;
        for (int i = 0; i < str.Length; i++)
        {
            if (str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u')
                vowels++;
        }
        if (vowels > maxvowels)
        {
            maxvowels = vowels;
            result = str;
        }
    }
    return result;
 }
}
```

147. Travel Agency

A travel agency has set standard tariffs for their pick up - drop services in a particular route.The route covers A,B,C,D locations one after the other. A. Tariff for the travel from Location A to Location B is 10 units/Km B. Tariff for the travel from Location B to Location C is 20 units/Km C. Tariff for the travel from Location C to Location D is 40 units/Km Return journey service is also provided. The starting point, destination point and the Time of travel (Normal - N, Untime - U) covered by a vehicle in a day are given as input1 in the format {XYZ...} - here X represents Start point , Y represents the destination point and Z represents the Time of travel. For untime travel,20% additional charges are applicable on actual tariff for that route. Write a program to calculate the total tariff collected by that vehicle for the day given and print the output in the following format, The car has taken A trips and has collected total amount of C

rupees. -Here A refers to the total number of services provided per day and C refers to the total amount from all the travels. Business rules: 1.If start point or destination points are invalid (other than A,B,C,D), print 'Invalid Location'. 2.If Time of travel is not either N or U , print 'Invalid Time of Travel'.

Create a class named UserProgramCode that has the following static method

public static int getTariffAmount(string[] input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the string array.

The next 'n' lines of input consists of strings that correspond to elements in the string array.

Refer business rules and sample output for output format.

Always display the tariff to be paid as an int.

Sample Input 1 : 4 ACN DAU ADN DCU

Sample Output 1 : The car has taken 4 trips and has collected total amount of 232 rupees Sample Input 2 :

4 ACN FAU ADN DCU

Sample Output 2 : Invalid Location

```
using System;

using System.Text.RegularExpressions;

namespace code1

{

  class Program

{


static void Main(String[] args)
```

```
{
  int n, amount;
n = int.Parse(Console.ReadLine());
  String[] input1=new String[n];

for (int i = 0; i < n; i++)
{
  input1[i] = Console.ReadLine();
}
  amount=UserMainCode.getTariffAmount(input1);
  if(amount!=-1&& amount!=-2)
  Console.WriteLine("The car has taken "+n+" trips and has collected total amount of " + amount + "
rupees");

}
}

}



using System;
```

```
public class UserMainCode
{
    public static int getTariffAmount(string[] input1)
{
  int length = input1.Length;
  double amount = 0;
  for (int i = 0; i <length;i++)
{
    if (input1[i][2] == 'N')
{
  if (input1[i][0] == 'A')
{
  if (input1[i][1] == 'B')
  amount += 10;
  else if (input1[i][1] == 'C')
  amount += 30;
  else if (input1[i][1] == 'D')
  amount += 70;
```

```
}
  else if (input1[i][0] == 'B')
{
  if (input1[i][1] == 'A')
                    amount += 10;
  else if (input1[i][1] == 'C')
  amount += 20;
  else if (input1[i][1] == 'D')
  amount += 60;
  else
{
  Console.WriteLine("Invalid Location"); return -1;
}
}
  else if (input1[i][0] == 'C')
{
  if (input1[i][1] == 'A')
  amount += 30;
  else if (input1[i][1] == 'B')
  amount += 20;
  else if (input1[i][1] == 'D')
  amount += 40;
  else
```

```
{
  Console.WriteLine("Invalid Location"); return -1;
}
}
  else if (input1[i][0] == 'D')
{
  if (input1[i][1] == 'A')
  amount += 70;
  else if (input1[i][1] == 'B')
  amount += 60;
  else if (input1[i][1] == 'C')
  amount += 40;
  else
{
  Console.WriteLine("Invalid Location"); return -1;
}
}
  else
{
  Console.WriteLine("Invalid Location"); return -1;
}
}
  else if (input1[i][2] == 'U')
```

```
{
  if (input1[i][0] == 'A')
{
  if (input1[i][1] == 'B')
  amount += 10 * 1.2;
  else if (input1[i][1] == 'C')
  amount += 30 * 1.2;
  else if (input1[i][1] == 'C')
  amount += 70 * 1.2;
}
  else if (input1[i][0] == 'B')
{
  if (input1[i][1] == 'A')
  amount += 10 * 1.2;
  else if (input1[i][1] == 'C')
  amount += 20 * 1.2;
  else if (input1[i][1] == 'D')
  amount += 60 * 1.2;
  else
{
  Console.WriteLine("Invalid Location"); return -1;
}
}
```

```
  else if (input1[i][0] == 'C')
{
  if (input1[i][1] == 'A')
              amount += 30 * 1.2;
  else if (input1[i][1] == 'B')
  amount += 20 * 1.2;
  else if (input1[i][1] == 'D')
  amount += 40 * 1.2;
  else
           {
  Console.WriteLine("Invalid Location"); return -1;
}
}
  else if (input1[i][0] == 'D')
{
  if (input1[i][1] == 'A')
  amount += 70 * 1.2;
  else if (input1[i][1] == 'B')
  amount += 60 * 1.2;
  else if (input1[i][1] == 'C')
  amount += 40 * 1.2;
  else
{
```

```
  Console.WriteLine("Invalid Location"); return -1;
}
}
  else
{
  Console.WriteLine("Invalid Location"); return -1;
      }




        }
  else
  { Console.WriteLine("Invalid Time of Travel"); return -2; }
}
  return (int)amount;
}
}
```

148. Add Days

Write a program which can print the date n days after the given date.

The date should be given in string format “mm/dd/yyyy” without time and the resultant added date should also be in the format “mm/dd/yyyy”.

Only positive value should be given as input to the days to be added, else print “n value is negative”. If the date format is not “mm/dd/yyyy” , then print “Invalid date format” .

Example : 5 days after “10/21/2009” is “10/26/2009”.

Include a class UserProgramCode with a static method addDays which accepts a String and an Integer. The return type (String) should return the final date as String or it would return "-1" if the day value is negative or it would return "-2" if the date is not as per the given format.

Create a Class Program which would be used to accept a String, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a String and an integer, where the String corresponds to the input date and the integer corresponds to the number of days.

Output consists of a String.

Refer sample output for formatting specifications.

Sample Input 1:

10/21/2009

5

Sample Output 1:

10/26/2009

Sample Input 2:

10/21/2009

-5

Sample Output 2:

n value is negative

Sample Input 3:

40/21/2009

5

Sample Output 3:

Invalid date format

```
class Program
  {
    static void Main(string[] args)
    {
      string s=Console.ReadLine();
      int i=Convert.ToInt16(Console.ReadLine());
      string s1 = UserProgramCode.addDays(s,i);
      if (s1 == "-1")
        Console.WriteLine("n value is negative");
      else if (s1 == "-2")
        Console.WriteLine("Invalid date format");
      Console.WriteLine(s1);

      Console.ReadLine();
    }
  }
  class UserProgramCode
  {
    public static string addDays(string s,int a)
    {
      string format = "MM/dd/yyyy";
```

```
        DateTime dt;
        bool b=DateTime.TryParseExact(s,format,null,System.Globalization.DateTimeStyles.None,out dt);
        if (!b)
           return "-2";
        if (a < 0)
           return "-1";
        dt=dt.AddDays(a);

        return dt.ToString("MM/dd/yyyy");
     }
  }
```

149. Check Anagrams

A anagram is a word or a phrase that can be created by rearranging the letters of another given word or phrase. We ignore white spaces and letter case. All letters of 'Desperation' can be rearranged to the phrase 'A Rope Ends It'. Write a program to check whether the 2 given strings are anagrams or not. Business Rules : 1. If there are any special characters (Space is not considered as special character) in either of the input strings, then store FALSE in the output variable. Create a class named UserProgramCode that has the following static method

public static bool checkAnagram(string input1,string input2) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of 2 strings.

Output is either “TRUE” or “FALSE”.

Refer business rules and sample output for formatting specifications. Sample Input 1 : tea eat

Sample Output 1 :

TRUE Sample Input 2 : Desperation A Rope Ends It

Sample Output 2 :

TRUE

```
using System;


class Program
{
  public static void Main( string[] args )
  {
            string str1 = Console.ReadLine();
            string str2 = Console.ReadLine();
    Console.WriteLine(UserProgramCode.checkAnagram(str1, str2).ToString().ToUpper());
  }
}
```

```
using System;


class UserProgramCode
{
  public static bool checkAnagram(string str1, string str2)
  {
            // fill your code here
     str1 = str1.ToLower();
     str1 = str1.Replace(" ", "");
     str2 = str2.ToLower();
```

```
str2 = str2.Replace(" ", "");


foreach (char item in str1)
{
   if (!char.IsLetter(item))
   {
      return false;
   }
}
foreach (char item in str2)
{
   if (!char.IsLetter(item))
   {
      return false;
   }
}


foreach (char c in str1)
{
   int ix = str2.IndexOf(c);


   if (ix == -1)
      return false;
```

```
    }

    return true;
  }
}
```

150. GCD – Array

Given an array of integers as input, write a program to find the Greatest Common Divisor for all the integer elements present in the input. Greatest Common Divisor also known as the greatest common factor (gcf), of two or more integers is the largest positive integer that divides the numbers without a remainder. Business Rule: 1. If the input array contains any value less than 1, assign -1 to the output1 variable. Create a class named UserProgramCode that has the following static method

public static int greatestCommonDivisor(int[] input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.

The next 'n' lines of input consist of elements in the input array.

Output is an integer.

Refer business rules and sample output for formatting specifications.

Sample Input 1 : 4 24 12 20 8

Sample Output 1 :

4 Sample Input 2 : 4 2 4 8 -6

Sample Output 2 :

-1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace gcd
{
   class UserProgramCode
   {
      public static int greatestCommonDivisor(int[] input1)
      {
         int count = 1;
         int[] ni = new int[67];
         int[] na = new int[67];
         int f = 0;


         int k = 0;
         for (int i = 0; i < input1.Length; i++)
         {
            if (input1[i] < 0)
            {
               return -1;
            }
```

```
}


for (int i = 0; i < input1.Length; i++)
{
   for (int j = 1; j <= input1[i]; j++)
   {
      if (input1[i] % j == 0)
      {
         ni[k] = j;
         k++;
      }
   }
}
Array.Resize(ref ni, k);
for (int i = 0; i < k; i++)
{
   count = 1;
   for (int j = i + 1; j < k; j++)
   {
      if (ni[i] == ni[j])
      {
         count++;
```

```
            }



        }



        if (count == input1.Length)
        {


            na[f] = ni[i];
            f++;
        }
      }
      int ans = na.Max();
      return ans;


    }
  }
}




using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace gcd
{
   class Program
   {
      static void Main(string[] args)
      {
         int n = Convert.ToInt32(Console.ReadLine());
         int[] nn = new int[n];
         for (int i = 0; i < n; i++)
         {
            nn[i] = Convert.ToInt32(Console.ReadLine());
         }
         int op = UserProgramCode.greatestCommonDivisor(nn);
         Console.WriteLine(op);
         Console.ReadLine();
      }
   }
}
```

151. Remove Duplicates

Given a string as input, write a program to remove duplicate characters from the string. Note - Only the first occurrence of each character should be retained. Retain all blank spaces. Business Rule: If there is no duplicate found, print -1.

Create a class named UserProgramCode that has the following static method

public static string removeDuplicates(string input1) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of a string.

Refer business rules and sample output for formatting specifications.

Sample Input :

hi this is sample test

Sample Output : hi ts ample

```
using System;
using System.Linq;
using System.Collections.Generic;
using System.Text;


   class Program
   {
      public static void Main(string[] args)
      {
         string input;
         input = Console.ReadLine();
         string outputStr = UserProgramCode.removeDuplicates(input);
         Console.WriteLine(outputStr);
```

```
        Console.ReadLine();
      }
   }
##############################################
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


class UserProgramCode
{
   public static string removeDuplicates(string input1)
   {

      //char[] ch = input1.ToCharArray();
      //char[] res = ch.Distinct().ToArray();+
      int i = 0,j=0;
      //int len = input1.Length;
      //char c='*';
      for (i = 0; i < input1.Length; i++)
      {
         for (j = i + 1; j < input1.Length; j++)
         {
            if (!(input1[j] == ' '))
            {
               if (input1[i] == input1[j])
               {
                  input1 = input1.Remove(j, 1);
               }
            }
         }
```

```
        }
        //input1.Remove(4, 1);

        string output = input1;
        //Console.Read();
        return output;



    }
}
```

152.Sum Of Squares Of Even Digits

Write a program to read a positive integer and to calculate the sum of squares of even digits available in the given number. Print the output.

Example:

input = 56895

output = 6*6 + 8*8 = 100

Include a class UserProgramCode with a static method sumOfSquaresOfEvenDigits which accepts an Integer. The return type (Integer) should return the sum of squares of even digits available in the given number.

Create a Class Program which would be used to accept an Integer, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of an Integer.

Output consists of an Integer, the sum of squares of even digits available in the given number .

Refer sample output for formatting specifications.

Sample Input 1:

56895

Sample Output 1:

100

```
class Program
  {
    static void Main(string[] args)
    {

      int i=Convert.ToInt32(Console.ReadLine());
      int s1 = UserProgramCode.sumOfSquareofEvenDigits(i);

      Console.WriteLine(s1);

      Console.ReadLine();
    }
  }
  class UserProgramCode
  {
    public static int sumOfSquareofEvenDigits(int a)
    {
```

```
        int sum=0;
        string a1 = a.ToString();
        int a2 = 0;
        for (int i = 0; i < a1.Length; i++)
        {
           a2 = (int)(a1[i])-48;
           if (a2 % 2 == 0)
           {

              sum = sum + (a2 * a2);

           }
        }
        return sum;
     }
  }
```

153 .Validate ID Locations

Given a method with two string inputs. First string is ID and second string is location. ID is in the format CTS-LLL-XXXX where LLL is the first three letters of given location and XXXX is a four digit number. If the given ID meets the given format return 1 else return -1 . Example: Input1 = CTS-hyd-1234 Input2 = hyderabad output = 1

Create a class named UserProgramCode that has the following static method

public static int validateIDLocations(string input1, string input2)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:

Input consists of 2 strings.

Output is an integer.

Sample Input : CTS-hyd-1234

hyderabad

Sample Output :

1

```
 using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace idValidation
{
   class Program
   {
      static void Main(string[] args)
      {
         string s1 = Console.ReadLine();
         string s2 = Console.ReadLine();
        int i = UserProgramCode.validateIDlocations(s1, s2);
         if (i == 1)
         {
            Console.WriteLine("valid");
         }
         else
         {
            Console.WriteLine("Invalid");
         }
         Console.ReadLine();
      }
   }
```

```
}



using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;

namespace idValidation
{
   class UserProgramCode
   {
      public static int validateIDlocations(string s1, string s2)
      {
         int output = 0;
         Regex reg = new Regex(@"^([CTS]+[-]+([A-Za-z]{3})+[-]+([0-9]{4}))$");
         if (reg.IsMatch(s1))
         {
            string res = s2.ToLower();
            if (s1.Contains(res.Substring(0, 3)))
            {
               output = 1;
            }
            else
            {
               output = -1;
            }
         }
         else
         {
```

```
            output = -2;
          }
          return output;
       }
    }
}
```

154.Vowels

A string is said to be valid if it contains exactly five vowels in any order. Assume there is no repetition of any vowel in the given string.

Example:
Input : acbisouzze
Output: Valid

Include a class UserProgramCode with a static method testVowels that accepts a string and returns an integer. The method returns 1 if the string is valid. Else it returns -1.

Create a class Program which would get the input and call the static method testVowels() present in the UserProgramCode.

If there are exactly five vowels present in the string then print "Valid" else print as "Invalid".
Sample Input 1: education Sample Output 1: Valid Sample Input 2: vowels Sample Output 2: Invalid

Vowels

```
   using System;
using System.Text;
using System.Linq;
```

```
public class UserMainCode
{
   public static int testVowels(string str)
   {
      if ((str.Contains('a') || str.Contains('A')) && (str.Contains('e') || str.Contains('E')) &&
(str.Contains('i') || str.Contains('I')) && (str.Contains('o') || str.Contains('O')) && (str.Contains('u') ||
str.Contains('U')))

      {
         return 1;
      }
      else
      {
         return -1;
      }
   }
}
using System;

public class Program {
   public static void Main(){
      string st = Console.ReadLine();
      int result = UserMainCode.testVowels(st);
      if (result == 1)
         Console.WriteLine("Valid");
      else
         Console.WriteLine("Invalid");
      Console.Read();
   }
}
```

155.Count the number of Occurrences

Write code to read two lines(String) and count the number of occurrences of second word of second sentence in the first sentence. Print the count.

Note - Consider case.

Example:

Input1: Hi this is cognizant Academy

Input2: Hello this is a trainee

Output: 1

Business Rule:

If there is no occurrence of the second word of second sentence with respect to the first sentence, return 0.

Include a class UserProgramCode with a static method countNoOfWords which accepts two strings. The return type (integer) should return the count.

Create a Class Program which would be used to accept two strings and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of two strings.

Output consists of an integer (the count).

Refer sample output for formatting specifications.

Sample Input 1:

Hi this is cognizant Academy

Hello this is a trainee

Sample Output 1:

1

m using System;

```
class UserProgramCode

{

  public static int countNoOfWords(string str1,string str2)

  {

            string[] sub1=str1.Split(' ');

    string[] sub2 = str2.Split(' ');

      int ctr=0;



    string str = sub2[1];



    for (int i = 0; i < sub1.Length; i++)

    {

       if (sub1[i] == str)

       {

          ctr++;

       }
```

```
    }

    return ctr;



  }

}



using System;

class Program

{

  public static void Main( string[] args )

  {

            string str1=Console.ReadLine();

            string str2=Console.ReadLine();

            Console.WriteLine(UserProgramCode.countNoOfWords(str1,str2));

    Console.ReadLine();
```

```
 }

}
```

156.Index power array

Write code to read an integer array and to find the power of each individual element according to its position index, add them up and print as output.

Example :

input = {7,6,2,1}

output = (7 power 0)+(6 power 1)+(2 power 2)+(1 power 3) = 1+6+4+1=12

Include a class UserProgramCode with a static method getSumOfPower which accepts an integer that corresponds to the size of the array and an integer array. The return type (Integer) should return the final output.

Create a Class Program which would be used to accept Input array and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of n+1 integers, where the first integer corresponds to the number of elements, followed by the array elements.

Output consists of an Integer(final output).

Refer sample output for formatting specifications.

Sample Input 1:

4

7

6

2

1

Sample Output 1:

12

6)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Reflection;

namespace ConsoleApplication2
{
   class Program
   {
      static void Main(string[] args)
      {
         int n = int.Parse(Console.ReadLine());
         if (n > 0)
         {
            int[] a = new int[n];
            for(int i=0;i<n;i++)
            {
               a[i] = int.Parse(Console.ReadLine());
            }
            int sum = UserProgramCode.getSumOfPower(n, a);
            Console.WriteLine(sum);
         }


      }
   }

}
```

```
class UserProgramCode
  {
     public static int getSumOfPower(int size,int[] a)
     {
        int sum=0;
        for(int i=0;i<size; i++)
        {
           sum+=(int)Math.Pow(a[i],i);

        }
        return sum;
     }
  }
```

157.Find nth Largest Number

Write a method to find the nth largest number in an input integer array. Include a class UserProgramCode with a static method findNthLargestNumber which accepts 2 inputs, an integer array and an integer (n) and returns an integer. If the input consists of any negative numbers, the method returns -1. Else the method returns the nth largest element in the array.

Create a class Program which would get the input and call the static method findNthLargestNumber present in the UserProgramCode. If the method returns -1, then print 'Invalid Input'.

Input and Output Format: The first line of the input consists of an integer that corresponds to m, the size of the array. The next m lines of input consists of integers that correspond to the elements in the array. The next line of input consists of integer that corresponds to 'n'. Refer sample output for formatting specifications.

Sample Input 1: 7 2 1 67 10 55 12 7 -2

Sample Output 1: Invalid Input Sample Input 2: 7 100 300 150 450 650 50 25 4

Sample Output 2: 150

2>>

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace TestPractice
{
   public class UserMainCode
   {

      public static int output1;
      public static int nthLargest(int[] input1, int input2)
      {
         foreach (var item in input1)
         {
            if (item < 0)
            {
               output1 = -1;
               return output1;

            }
         }
         if (output1 != -1)
         {
            Array.Sort(input1);
            Array.Reverse(input1);
            input1 = input1.Distinct().ToArray();
            output1 = input1[input2 - 1];
         }
         return output1;
```

```
    }
  }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace TestPractice
{
  class Program
  {
    static void Main(string[] args)
    {
      int n;
      n = Convert.ToInt32(Console.ReadLine());
      int[] input1 = new int[n];
      for(int i=0;i<n;i++)
      {
        input1[i]=Convert.ToInt32(Console.ReadLine());
      }
      int input2=Convert.ToInt32(Console.ReadLine());
      int res=UserMainCode.nthLargest(input1,input2);
      if(res==-1)
      {
        Console.WriteLine("invalid input");
      }
```

```
        else
        {
           Console.WriteLine(res);
        }
        Console.ReadLine();
     }
  }
}
```

158.Validate Phone Number

Write a program to read a phone number as a string input and to verify the phone number using following business rules:

-it should contain only numbers or dashes (-)

-dashes may appear at any position

-should have exactly 10 digits

If the Phone number is valid print “Valid” otherwise print “Invalid”.

Example:

input = 265-265-7777

output = Valid

Include a class UserProgramCode with a static method validatePhoneNumber which accepts a String. The return type (Integer) should return 1 if valid, else return 2.

Create a Class Program which would be used to accept a String, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a String, which corresponds to the phone number.

Output consists of a String, “Valid” if the phone number is valid, else “Invalid”.

Refer sample output for formatting specifications.

Sample Input 1:

265-265-7777

Sample Output 1:

Valid

Sample Input 2:

1111-111-1111

Sample Output 2:

Invalid

47.Validate phone number

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication18
{
   class UserProgramCode
   {
      public static int validatephonenumber(string s)
      {
         int len = s.Length;
         int digit = 0, flag = 0;
         char[] a = s.ToCharArray();
         for (int i = 0; i < len; i++)
         {
            if ((a[i] == '-') || char.IsDigit(a[i]))
               flag++;
         }
```

```
        if (flag == len)
        {

            for (int i = 0; i < len; i++)
            {
               if (char.IsNumber(a[i]))
               {
                  digit++;
               }

            }

        }
        if (digit == 10)
           return 1;
        else

           return 2;


      }
   }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication18
{
```

```
  class Program
  {
     static void Main(string[] args)
     {
        UserProgramCode u = new UserProgramCode();
        string s;
        int result;
        s = Console.ReadLine();
        result = UserProgramCode.validatephonenumber(s);
        if(result==1)
           Console.WriteLine("valid");
        else if(result==2)
           Console.WriteLine("Invalid");
     }

  }
}
```

==========

S

159.Get Big Diff

Write a program that reads an integer array and finds the difference between the largest element and smallest element in the array.

Include a class UserProgramCode with a static method getBigDiff that accepts an integer array and returns an integer.

Create a Class Program which would be used to read the integer array and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of n+1 integers. The first integer corresponds to n, the number of elements in the array.
The next 'n' integers correspond to the elements in the array.
Output consists of an integer.

Sample Input 1: 4 10 3 5 6 Sample Output 1: 7 Sample Input 2: 4 2 -10 7 -2 Sample Output 2: 17

50.Get Big Diff

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication18
{
  class Program
  {
     static void Main(string[] args)
     {
        UserProgramCode u=new UserProgramCode();
        int n = int.Parse(Console.ReadLine());
        int[] a = new int[n];
        int result;
        for (int i = 0; i < n; i++)
           a[i] = int.Parse(Console.ReadLine());
        result = UserProgramCode.getBigDiff(a);
        Console.WriteLine(result);
     }

  }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication18
{
   class UserProgramCode
   {
      public static int getBigDiff(int[] a)
      {
        int  min = a[0],max=a[0],diff=0;
        for (int i = 0; i < a.Length; i++)
        {
            if (a[i] >= max)
                max = a[i];
            if (a[i] <= min)
                min = a[i];
        }
        diff= max - min;
        return diff;
      }
   }
}
```

160.Sum Common Elements

Write a program to read two int arrays, eg. A{2,3,5,1} and B{1,3,9}, and to find out sum of common elements in given arrays. Print the sum, or print “No common elements found” if there are no common elements.

Assume the common element appears only once in each array.

Include a class UserProgramCode with a static method getSumOfIntersection which accept the size of two integer arrays and the two integer arrays. The return type (integer) should return the sum, or -1, accordingly.

Create a Class Program which would be used to accept two integer arrays, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of n+m+2 integers, where first two integers corresponds to the size of the two array lists, respectively, followed by the corresponding array elements.

Output consists of an Integer(the corresponding output) or string - ("No common elements found").

Refer sample output for formatting specifications.

Sample Input 1:

4

3

2

3

5

1

1

3

9

Sample Output 1:

4

Sample Input 2:

4

3

2

31

5

14

1

3

9

Sample Output 2:

No common elements found

18)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Fwd_Prgs
{
   public class UserProgramCode
   {
      public static int getSumOfIntersection(int n1, int n2, int[] a, int[] b)
      {
         int sum=0;
         for (int i = 0; i < n1; i++)
         {
            for (int j = 0; j < n2; j++)
               if(a[i]==b[j])
                  sum = sum + a[i];

         }
         if (sum == 0)
            return -1;
         else
```

```
            return sum;
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            int n1 = int.Parse(Console.ReadLine());
            int n2 = int.Parse(Console.ReadLine());
            int[] a=new int[n1];
            int[] b=new int[n2];
            for(int i = 0; i < n1; i++)
              a[i] = int.Parse(Console.ReadLine());
            for(int i = 0; i  <n2; i++)
              b[i] = int.Parse(Console.ReadLine());
            int res = UserProgramCode.getSumOfIntersection(n1, n2, a, b);
            if(res==-1)
               Console.WriteLine("No common elements found");
            else
               Console.WriteLine(res);
        }
    }
}
```

161.Quadratic Equation

Consider two equations x^2 - 2y + z = 0 and x^2+y=40 . Given an input integer array input1 containing values for the variable 'x', write a program to apply the input values to the equations and find out the corresponding y and z values to store them in the output in the following format (y1,z1,y2,z2,....} and so on. Print the output array Output array elements can contain negative values. Business rule: 1) If any of

the elements in input1 array is negative, then print -1. 2) If there are any duplicates found in input1 array, then print -2. 3) If size of the input1 array is 1 or greater than 10, then print -3. Example 1: input1 : {1,2,4,5} output1: {39,77,36,68,24,32,15,5} Example 2: input1 : {5,8,3,-4,6} output1: {-1}

Create a class named UserProgramCode that has the following static method

public static int[] quadEquation(int[] input1) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.

The next 'n' lines of input consist of elements in the input array.

Refer business rules and sample output for formatting specifications. Sample Input 1 :

4

1

2

4

5

Sample Output 1 :

39

77

36

68

24

32

15

5 Sample Input 2 : 5

5

8

3

-4

6

Sample Output 2 :

-1

```
 using System;
using System.Collections.Generic; using System.Linq; using System.Text;

namespace ConsoleApplication9
{    class Program
   {       static void Main(string[] args)
      {
         int[] arr=new int[5];
         Console.WriteLine("Enter the numbers of the array");
          for (int i = 0; i < arr.Length; i++)
         {
            arr[i] = Convert.ToInt32(Console.ReadLine());
         }
         int[] result=sourav(arr);
         for (int i = 0; i < result.Length; i++)
   {
            Console.WriteLine(result[i]+",");
         }
         Console.ReadLine();
       }
      public static int[] sourav(int[] arr)
      {
         List<int> li = new List<int>();          int[] arr1 = new int[]{-1};          int[] arr2 = new int[] { -2 };
List<int> li1 = new List<int>();

         for (int i = 0; i < arr.Length; i++)
         {
            if (arr[i] < 0)
            {
```

```
            return arr1;
          }

        }

        for (int i = 0; i < arr.Length; i++)
        {
          if (!li1.Contains(arr[i]))
          {
            li1.Add(arr[i]);
          }            else            {
            return arr2;
          }
        }

        int y = 0,z=0;
        for (int i = 0; i < arr.Length; i++)
        {
y         = (int)(40 - Math.Pow(arr[i], 2));            li.Add(y);
z         = (int)((2 * y) - Math.Pow(arr[i], 2));            li.Add(z);
        }          return li.ToArray();
      }
   }
}
```

Method 2:

```
using System;
using System.Collections.Generic;
using System.Linq;
```

```
using System.Text;

namespace quadratic
{
   class Program
   {
      static void Main(string[] args)
      {
         int n = int.Parse(Console.ReadLine());
         int[] xarr = new int[n];
         for (int i = 0; i < n; i++)
         {
            xarr[i] = int.Parse(Console.ReadLine());
         }
         int[] op = UserProgramCode.quadEquation(xarr);
         int len = op.Length;

         if (op[0] == -1)
         {
            Console.WriteLine("-1");
         }
         else if (op[0] == -2)
         {
            Console.WriteLine("-2");
         }
         else if (op[0] == -3)
         {
            Console.WriteLine("-3");
         }
         else
            for (int i = 0; i < op[len - 1]; i++)
            {
```

```
            Console.WriteLine(op[i]);
        }
        Console.ReadLine();

    }
  }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace quadratic
{
  class UserProgramCode
  {
    public static int[] quadEquation(int[] input1)
    {
      int z= 0;
      int y = 0;
      int k = 0;
      int flag = 1;
      int[] temp = new int[30];
      for (int i = 0; i < input1.Length; i++)
      {
        if (input1[i] < 0)
        {
          temp[0] = -1;
```

```
        flag = 0;


    }
}
if (flag == 1)
{
    if (input1.Length < 2 || input1.Length > 10)
    {
        temp[0] = -3;
        flag = 0;


    }
}
if (flag == 1)
{
    for (int i = 0; i < input1.Length; i++)
    {
        for (int j = i + 1; j < input1.Length; j++)
        {
            if (input1[i] == input1[j])
            {
                temp[0] = -2;
                flag = 0;


            }
        }
    }
}


if (flag == 1)
{
    for (int i = 0; i < input1.Length; i++)
```

```
            {
                z = 0;
                y = 0;
                y = 40 - (input1[i] * input1[i]);
                z = (2 * y) - (input1[i] * input1[i]);
                temp[k] = y;
                temp[k + 1] = z;
                k = k + 2;


            }
        }
        if(flag==1)
        temp[29] = k;
        return temp;
    }
  }
}
```

162. Number System

Given a number system having numbers which is a combination of digits 3 and 4 only. First few numbers in the number system are: 3, 4, 33, 34, 43, 44, 333, 334, 343, 344, 433, 434, 443, 444, 3333, 3334, 3343, 3344, 3433, 3434, 3443, 3444, … Find the nth number in the number system where n is an integer index given as input . Business Rule: 1. If the input1 is less than 1, then print -1 Create a class named UserProgramCode that has the following static method

public static void findNumber(int input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:

Input is an integer.

Output is an integer.

Sample Input 1 : 10

Sample Output1 :

344

Sample Input 2 :

-8

Sample Output 2 :

-1

163.File Extension

Write a program to read a file name as a string and find out the file extension and return it as output. For example, the file sun.gif has the extension gif.

Include a class UserProgramCode with a static method fileIdentifier which accepts a string. The return type (string) should return the extension of the input string (filename).

Create a Class Program which would be used to accept Input String and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a string that corresponds to a file name.

Output consists of a string(extension of the input string (filename)).

Refer sample output for formatting specifications.

Sample Input 1:

sun.gif

Sample Output 1:

gif

```
class Program
{
   static void Main(string[] args)
   {
      string s;
```

```
        s = Console.ReadLine();
        userprogramcode obj = new userprogramcode();
        Console.WriteLine(obj.fileIdentifier(s));



    }
}
public class userprogramcode
{
    public string fileIdentifier(string s)
    {
        string[] str;
        str=s.Split('.');
        return str[1];
    }
}
```

164.Calculate Bill Amount

A Electrical shop has announced the following seasonal discount for the purchase of certain items. IInclude a class UserProgramCode with a static method calculateBillAmount which accepts double and char value as input and returns an integer that corresponds to the net amount. Compute the net amount to be paid by the customer based on the below criteria, Purchase Amount(Rs) Discount on Tv Discount on MusicSystem 1-25000 5% 10% 25001-50000 10% 20% More than 50000 15% 30% [Hint: DiscountPrice=(DiscountRate/100)*Amount of Purchase. Net Amount=Amount of Purchase-DiscountPrice.] If the purchase item is other than TV and MusicSystem return -2, if the purchase amount is a negative value return -1.Otherwise return the net amount. Create a Main class which gets double and char as an input and call the static method calculateBillAmount present in the UserProgramCode. If the method returns -1, then print 'Negative Values'. If the method returns -2, then print 'No Items'.
Input and output format: The input will be a double and char values. If the input character is 'T', it corresponds to TV. If the input character is 'M', it corresponds to Music System.
Refer sample output for formatting specifications.
Sample Input 1: 20000 X

Sample Output 2: No Items

Sample Input 2: -5000 M

Sample Output 2: Negative Values

Sample Input 3: 70000 T

Sample Output 3: 59500

calculate bill amount

```
using System;

  class Program
  {
    public static void Main( string[] args )
    {
                    double input1;
                    char input2;
                    input1 = Convert.ToDouble(Console.ReadLine());
                    input2 = Convert.ToChar(Console.ReadLine());
                    int value = UserProgramCode.calculateBillAmount(input1,input2);
                    if(value == -2)
                                Console.Write("No Items");
                    else if(value == -1)
                                Console.Write("Negative Values");
                    else
                                Console.Write(value);
      Console.ReadLine();

    }
  }
```

```
using System;

  class UserProgramCode
  {
     public static int calculateBillAmount(double input1, char input2)
     {
        double dp;
        double na = 0;
        char[] ch = new char[] { 'T', 'M' };

        if (input1 < 0)
        {
           return -1;
        }
        if ((input2 != 'T') && input2 != 'M')
        {

           return -2;
        }




        if (input1 >= 1 && input1 <= 25000)
        {
           if (input2 == 'T')
           {
              dp = (0.05 * input1);
              na = (input1 - dp);

              return (int)na;
           }
```

```
    else if (input2 == 'M')
    {
        dp = (0.1 * input1);
        na = (input1 - dp);
        return (int)na;
    }
    else
    {
        Console.WriteLine("not valid");
    }
}
else if (input1 >= 25001 && input1 <= 5000)
{
    if (input2 == 'T')
    {
        dp = (0.1 * input1);
        na = (input1 - dp);
        return (int)na;
    }
    else if (input2 == 'M')
    {
        dp = (0.2 * input1);
        na = (input1 - dp);
        return (int)na;
    }
    else
    {
        Console.WriteLine("not valid");
    }
}
if (input1 >= 50000)
{
```

```
            if (input2 == 'T')
            {
               dp = (0.15 * input1);
               na = (input1 - dp);
               return (int)na;
            }
            else if (input2 == 'M')
            {
               dp = (0.3 * input1);
               na = (input1 - dp);
               return (int)na;
            }
            else
            {
               Console.WriteLine("not valid");
            }
         }
         return (int)na;


      }
   }
```

165.Calculate Frequency

Given two string inputs input1 and input2, write a program to find the number of times the complete string in input1 occurs in input2 and print the count. Ignore case sensitiveness in the input strings. Business Rules: 1)If input1 has repeated words, print -1. 2)If the count of occurrence is zero then print -2 . Example1: input1: A good place input2: It is a good place to be in and a good place to have fun. output: 2 Example: input1 :Does he have to have a car ? input2: Yes he should. output: -1 Create a class named UserProgramCode that has the following static method

public static int calcFrequency(string input1, string input2)

Create a class named Program that accepts the input and calls the static method present in the

UserProgramCode. Input and Output Format: Input consists of 2 strings.

Output consists of an integer.

Sample Input : A good place It is a good place to be in and a good place to have fun.

Sample Output : 2

77.

```
//program.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace oops2
{


   class Program
   {
      static void Main(string[] args)
      {
         String input1 = Console.ReadLine();
         String input2 = Console.ReadLine();
         int ret = UserProgramCode.calcFrequency(input1, input2);
         Console.WriteLine(ret);
         //Console.WriteLine(input1.ToLower());
      }
   }

}
//UserProgramCode.cs
using System;
using System.Collections.Generic;
```

```
using System.Linq;
using System.Text;

namespace oops2
{
   class UserProgramCode
   {
      public static int calcFrequency(string input1, string input2)
      {
         StringBuilder sb = new StringBuilder();
         int val = 0;
         String[] arr = input1.Split(' ');
         foreach (String a in arr)
         {
            sb.Append(a.ToLower());
         }
         int len = arr.Length;
         for (int i = 0; i < len; i++)
         {
            String s = arr[i];
            for (int j = i + 1; j < len; j++)
            {
               if (s.Equals(arr[j]))
               {
                  val++;
                  return -1;
               }
            }
         }
         if (val == 0)
         {
            int count = 0;
```

```
 StringBuilder sb1 = new StringBuilder();
 String linput1 = input1.ToLower();
 String linput2 = input2.ToLower().Replace('.',' ');
// Console.WriteLine(linput2);
 String[] l = linput2.Split(' ');
 foreach (String a in l)
 {
    sb1.Append(a);
 }
 len = sb.Length;
 int len2 = sb1.Length;
// Console.WriteLine(len + "" + len2);
 for (int i = 0; i < len2; i++)
 {
    if(i<len2-len)
    {
      // Console.WriteLine(i+","+len);
      // Console.WriteLine(sb);
      // Console.WriteLine(sb1);
       String sub = sb1.ToString().Substring(i, len);
       if (sub.Equals(sb.ToString()))
       {
          count++;
       }
    }

 }
 if (count == 0)
    return -2;
 else
    return count;
```

```
        }
        return 0;
      }
   }
}
```

166.Power of 2

Write a program to check whether an integer number is a power of 2 or not. If it is a power of 2 print the power else print -1 . Business Rule: 1. If the given input integer is a negative number/not a power of 2, print -1. Create a class named UserProgramCode that has the following static method

public static int twoPower(int input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:

Input consists of an integer.

Output consists of an integer. Refer business rules and sample output for the format.

Sample Input 1 : 1024

Sample Output 1 : 10 Sample Input 2 : 6 Sample Output 2 :

-1

94.Power of 2

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Collections;
namespace ConsoleApplication23
{
   class UserProgramCode
   {
      public static int twoPower(int input1)
```

```
{
   int i1 = input1;
   int n=2;
   int i = 1;
   int sum=1;
   int val=0;
   ArrayList al = new ArrayList();
   while (i < i1)
   {
      if ((sum * n) == i1)
      {
         val = i;
         break;
      }
      sum = sum * n;
      al.Add(sum);
      i++;
   }

   return val;

}
static void Main(string[] args)
{
   int i;
   i = int.Parse(Console.ReadLine());
   int val1;
   val1 = twoPower(i);
   if (val1 > 0)
      Console.WriteLine(val1);
   else
      Console.WriteLine(-1);
```

```
    }
  }
}
```

167. Shortest Word Length

Given a string array as input, write a program to find the length of the shortest word in the array..

Create a class named UserProgramCode that has the following static method

public static int shortestWordLength(string[] input1)

Create a class named Program that accepts a string array as input and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer 'n' that corresponds to the number of elements in the string array.

The next 'n' lines of the input consists of strings that correspond to the elements in the string array.

Output consists of a single integer that corresponds to the length of the shortest word in the array.

Sample Input :

3 cherry apple blueberry Sample Output : 5

Program 72:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;


namespace ConsoleApplication18

{
```

```
class Program
{
   static void Main(string[] args)
   {
      int n = Convert.ToInt32(Console.ReadLine());
      string[] a = new string[n];
      for (int i = 0; i < a.Length; i++)
         a[i] = Console.ReadLine();
      int r = userProgramCode.shortestWordLength(a);
      Console.WriteLine(r);
      Console.ReadLine();
   }
}
class userProgramCode
{
   public static int shortestWordLength(string[] a)
   {
      int  min=1000;
      for (int i = 0; i < a.Length; i++)
      {
         if (i == 0)
            min = a[i].Length;
         else
```

```
            {
                if (min > a[i].Length)
                    min = a[i].Length;
            }
        }
        return min;
    }
  }
}
```

168. Calculate Simple Interest

Write a customized Simple Interest Calculator to calculate simple interest for any given inputs of Principal P and years N for a private bank in Europe.

Business Rules:

1) If principal is greater than or equal to 99999 & less than 500001 and years > 5, the interest rate is 8.75%.

2) If principal is greater than or equal to 500000 & less than 1000001 and years > 3, the interest rate is 9.25%.

3) Any other principal amount and years will attract the standard interest rate of 8.25%.

4) The bank can accept max of 1000000 for a deposit. For amount greater than 1000000 the calculator should return -1.

5) Round off the interest to 2 decimal places.

6) Simple interest formula: (P*N*R) /100.

7) All the inputs should be non negative values. Else return -1.

Include a class UserProgramCode with a static method calculateSimpleInterest which accepts two integers and returns a double. The first input parameter refers to the principal P and the second input parameter refers to the years N.

The return type (Double) should return Simple Interest amount. Refer Business Rules and return -1 accordingly.

Create a Class Program which would be used to accept two Integers, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of two integers, the principal amount and the number of years, respectively.

Output consists of an Integer (the gift voucher amount) or a String “Interest cannot be calculated” if any of the inputs is invalid.

Refer sample output for formatting specifications.

Sample Input 1:

100000

1

Sample Output 1:

8250

Sample Input 2:

10000000

1

Sample Output 2:

Interest cannot be calculated

simple interest

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Simple_Interest
```

```
{
   class Program
   {
      static void Main(string[] args)
      {
         int pin = int.Parse(Console.ReadLine());
         int year = int.Parse(Console.ReadLine());
         double op=UserProgramCode.calculatesimpleinterest(pin, year);


          if(op==-1)
            Console.WriteLine("Interest cannot be calculated");
          else

          Console.WriteLine(op);
         Console.ReadLine();
      }
   }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Simple_Interest
{
   class UserProgramCode
   {
      public static double calculatesimpleinterest(int pin, int year)
```

```
    {
        double final = 0;
        if (pin > 1000000)
            return -1;
        if (pin >= 99999 && pin <= 500001)
        {
            if (year > 5)
            {
                final = Math.Round((pin * year * 8.75) / 100, 2);
            }
        }
        if (pin <= 1000001 && pin >= 500001)
        {
            if (year == 3 || year == 4)
            {
                final = Math.Round((pin * year * 9.25) / 100, 2);
            }
        }
        if (year < 2 && pin < 1000000)
        {
            final = Math.Round((pin * year * 8.25) / 100, 2);
        }



        return final;
    }
  }
}
```

169. Sum of Odd Even Positioned

Write a program to find whether the sum of digits at even indexes and sum of digits at odd indexes in the given number are equal. Include a class UserProgramCode with a static method sumOfOddEvenPositioned that accepts an integer . The return type (integer) should return 1 if it

a valid, else return 2. Create a Class Program which would be used to read an integer and call the static method present in UserProgramCode. Input and Output Format:

Input consists of an integer.

Output consists of a String(“Valid” or “Not Valid”).

Refer sample output for formatting specifications.

Sample Input 1: 1221

Sample Output 1:

Valid

Sample Input 2:

2000204

Sample Output 2:

Not Valid

SUM OF ODD EVEN POSITIONED

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

class Program
{
   public static void Main(string[] args)
   {
      int input = Convert.ToInt32(Console.ReadLine());
      int value = UserMainCode.sumOfOddEvenPositioned(input);
      if(value == 1)
                              Console.WriteLine("Valid");
                  else
                              Console.WriteLine("Not Valid");
   }
}
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;

public class UserMainCode
{
   public static int sumOfOddEvenPositioned(int input)
   {
      int even = 0, odd = 0,i=1,j=0;
      while (input != 0)
      {
         j = input % 10;
         if (i % 2 == 0)
         {
            even = even + j;
         }
         else
         {
            odd = odd + j;
         }
         input = input / 10;
         i++;
      }
      if (even == odd)
         return 1;
      else
         return -1;
   }
}
```

170. Student Score

Given a string array Input (Input 1) containing Student name and percentage of marks in the below format Input = {StudentName1, Mark1, StudentName2, Mark2, StudentName3, Mark3,.....etc}, write a program to determine the student grade based on below condition, and print the output in the below format.

AAA has scored BBB marks with CCC scores

where AAA - Input StudentName (Input 2), BBB - Mark and CCC - Grade in Upper case. Grade calculation: If the mark is greater than or equal to 80, then OUTSTANDING If the mark is less than 80 and greater than or equal to 60, then GOOD If the mark is less than 60 and greater than or equal to 50, then AVERAGE If the mark is less than 50, then FAIL Business rule: 1) If any of the StudentName in Input1 or Input2 contains any special characters, then print “Invalid Input”. 2) If the Input2 string value is not present in Input1 array, then print “Invalid Student” 3) If the Input1 array length is odd, then print “No corresponding Student or Mark”

Create a class named UserProgramCode that has the following static method

public static string studentScore(string[] input1, string input2) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.

The next 'n' lines of input consist of elements in the input array.

The next line of the input consists of a string that corresponds to the student name.

Refer business rules and sample output for formatting specifications. Sample Input 1 : 4 Ram 55 Vignesh 89 Vignesh

Sample Output 1 : Vignesh has scored 89 marks with OUTSTANDING grade

Sample Input 2 : 5 Anil 76 Sunil 68 Raja Vignesh

Sample Output 2 : No corresponding Student or Mark

```
using System;

using System.Collections.Generic;
```

```
using System.Linq;
using System.Text;


namespace practice1
{
   class Program
   {
      static void Main(string[] args)
      {


         int size = Int32.Parse(Console.ReadLine());
         string[] str = new string[size];
         for (int i = 0; i < size; i++)
         {
            str[i] = Console.ReadLine();
         }
         string input2 = Console.ReadLine();
         Console.WriteLine(UserProgramCode.studentscore(str, input2));
         Console.ReadLine();



      }
   }
```

```
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace practice1
{
   class UserProgramCode
   {
      public static string studentscore(string[] s, string i2)
      {
         string[] str = s; string out1 = ""; int marks = 0, f1 = 0, f2 = 0; if (str.Length % 2 == 0)
         {
            for (int i = 0; i < str.Length; i++)
            {
               if (i % 2 == 0)
               {

                  foreach (char c in str[i])
                  {
```

```
    if (char.IsLetter(c))
    { f1 = 1; }
    else
    {
        f1 = -1;
    }
}
foreach (char c1 in i2.ToCharArray())
{
    if (char.IsLetter(c1))
    { f2 = 1; }
    else
    {
        f2 = -1;
    }
}
if (f1 == 1 && f2 == 1)
{
    if (s[i] == i2)
    {
        marks = Int32.Parse(str[i + 1]); if (marks >= 80)
            out1 = s[i] + " has scored " + str[i + 1] + " marks with OUTSTANDING grade";
        if (marks < 80 && marks >= 60)
```

```
                out1 = s[i] + " has scored " + str[i + 1] + " marks with GOOD grade";
            if (marks < 60 && marks >= 50)
                out1 = s[i] + " has scored " + str[i + 1] + " marks with AVERAGE grade";
            if (marks < 50) out1 = "fail";
          }
          else
          {
            out1 = "invalid student";
          }
        }
        else if (f1 == -1 || f2 == -1)
        {
          out1 = "invalid input";
        }
      }
    }


}
else
{
   out1 = "no marks or student";
}
```

```
        return out1;


    }


  }
}
```

171. Sum of Squares

Write a program to find the sum of the squares of first n natural numbers. If n less than 0, return -1.

Include a class UserProgramCode with a static method sumSquare which accepts an integer. The return type is an integer as given in the above statement.

Create a Class Program which would be used to accept Input and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of the value n.

Output consists of a integer as mentioned in the problem statement.

Refer sample output for formatting specifications.

Sample Input 1:

3

Sample Output 1:

14

Sample Input 2:

-5

Sample Output 2:

-1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Reflection;


namespace ConsoleApplication2
{
   class Program
   {
      static void Main(string[] args)
      {
         int n = int.Parse(Console.ReadLine());
         int result=UserProgramCode.sumSquare(n);
         if ( result== -1)
         {
            Console.WriteLine(-1);
         }
         else
         {
            Console.WriteLine(result);
         }
```

```
    }
  }

}

class UserProgramCode
  {
    public static int sumSquare(int n)
    {
      int sum = 0;
      if (n < 0)
        return -1;
      for (int i = 1; i <= n; i++)
      {
        sum += (int)Math.Pow(i, 2);
      }
      return sum;
    }
  }
```

172. Insurance Guide

An Insurance company follows the following rules to calculate premium.

(1) If a person's health is excellent and the person's age is in the range [25,35] (both 25 and 35 inclusive) and lives in a city and is a male then the premium is 4 Rs. per thousand and his policy amount cannot exceed Rs. 2 lakhs rupees.
(2) If a person satisfies all the above conditions except that the sex is female then the premium is 3 Rs. per thousand and her policy amount cannot exceed Rs. 1 lakh rupees.
(3) If a person's health is poor and the person's age is in the range [25,35] (both 25 and 35 inclusive) and lives in a village and is a male then the premium is 6 Rs. per thousand and his policy cannot exceed Rs. 10,000 rupees.
(4) In all other cases the person cannot be insured.

Write a program to display premium and maximum policy amount for given inputs.

Input1 - Health condition ('E' for excellent and 'P' for poor health)
input2 - Age
input3 - Gender('F' for female,'M' for Male)
input4 - Location('C' for City,'V' for Village)

Output1 - Premium per Thousand
Output2 - Maximum Insurance Amount the person can avail.

Example:

input1: E
input2: 30
input3: F
input4: C

Output1: 3
Outpu2: 100000

Business Rule:

1. If the person can't be insured then print “The person cannot be insured”.
2. If the person's age is more than 60 then print “Age limit Exceeded”.

Include a class UserProgramCode with a static method InsuranceGuide which accept three characters and an integer. The return type is void.

Create a Class Program which would be used to accept three characters and an integer , and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a character, which corresponds to the health condition, an Integer, which corresponds to the age, a character, which corresponds to the gender, a character, which corresponds to the location.

Refer sample output for formatting specifications.

Sample Input 1:

E

30

F

C

Sample Output 1:

3

100000

Sample Input 2:

E

70

F

C

Sample Output 2:

Age limit Exceeded

Sample Input 3:

P

50

F

V

Sample Output 3:

The person cannot be insured

insuranceguide

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

class Program
{
   public static void Main(string[] args)
   {
      char health = Convert.ToChar(Console.ReadLine());
      int age = Convert.ToInt32(Console.ReadLine());
      char gender = Convert.ToChar(Console.ReadLine());
      char location = Convert.ToChar(Console.ReadLine());
```

```
        UserProgramCode.InsuranceGuide(health, age, gender, location);
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

class UserProgramCode
{
    public static void InsuranceGuide(char health, int age, char gender, char location)
    {
        //Fill your code here
        if (health == 'E' && age >= 25 && age <= 35 && location == 'C' && gender == 'M')
        {
            Console.WriteLine(4);
            Console.WriteLine(200000);
        }
        else if (health == 'E' && age >= 25 && age <= 35 && location == 'C' && gender == 'F')
```

```
        {
            Console.WriteLine(3);
            Console.WriteLine(100000);
        }
        else if (health == 'P' && age >= 25 && age <= 35 && location == 'V' && gender == 'M')
        {
            Console.WriteLine(6);
            Console.WriteLine(10000);
        }
        else if(age>=60)
            Console.WriteLine("Age limit Exceeded");
        else
            Console.WriteLine("The person cannot be insured");
    }
}
```

173. Calculate Cost

Samira Florists take orders for flower decoration in social events and functions. They charge their customers based on:

The weight of flowers (kg) and
Decoration Options – Simple (S), Customized (C).
Flower type - Normal flowers (N) cost Rs.400 per kg

Exotic flowers (E) cost Rs. 700 per kg

They charge an additional Rs.15000 for simple decoration and Rs.25000 for a Customized decoration, apart from the cost of flowers. Also, they take only a minimum order of Rs. 20000 or more (including the flower cost and the decoration cost).

Business Rule:

If the cost calculated is less Rs 20000, then return -1.
If the type of flowers given is other than (N) or (E), then return -2.
If the type of decorations given is other than (S) or (C), then return -3.

Write a program to read an Integer and two characters, and calculate the total amount the customer pays to the florists. The values are: weight of flowers(in kg), the type of flowers (N or E)and the type of decoration(S or C). Print the final cost, or print “Too low cost”, if method returns -1, print “Invalid type of flower”, if method returns -2, print “Invalid decoration type”, if the method returns -3.

(Total Amount customer pays = Cost of flowers + Cost of Decoration)

Include a class UserProgramCode with a static method calculateCost which accept an integer and two characters . The return type (integer) should return output according to the business rules.
Create a Class Program which would be used to accept an integer and two characters, and call the static method present in UserProgramCode.
Input and Output Format:
Input consists of an integer, which corresponds to the weight, and two characters, which correspond to the flowers type and decoration type respectively.
Out put consists of an integer or a string.
Refer sample output for formatting specifications.

Sample Input 1:
10
N

B

Sample Output 1:

Invalid decoration type

Sample Input 2:

30

A

S

Sample Output 2:

Invalid type of flower

Sample Input 3:

10

N

S

Sample Output 3:

Too low cost

Sample Input 4:

20

N

S

Sample Output 4:

23000

24.CALCULATE COST

```
using System;



class Program
```

```
{
   public static void Main(string[] args)
   {
      int cost = Convert.ToInt32(Console.ReadLine());
      char fltype = Convert.ToChar(Console.ReadLine());
      char dctype = Convert.ToChar(Console.ReadLine());
      int result = UserProgramCode.calculateCost(cost, fltype, dctype);
      if (result == -1)
         Console.WriteLine("Too low cost");
      else if (result == -2)
         Console.WriteLine("Invalid type of flower");
      else if (result == -3)
         Console.WriteLine("Invalid decoration type");
      else
         Console.WriteLine(result);
   }
}
using System;


class UserProgramCode
{
   public static int calculateCost(int cost, char fltype, char dctype)
   {
```

```
        if (dctype != 'S' && dctype != 'C')
            return (-3);
        if (fltype != 'N' && fltype != 'E')
            return (-2);
        int value = 0;
        if (dctype == 'S')
            value = value + 15000;
        if (dctype == 'C')
            value = value + 25000;
        if (fltype == 'N')
            value = value + (cost * 400);
        if (fltype == 'E')
            value = value + (cost * 700);
        if (value < 20000)
            return -1;
        else
            return (value);

    }
}
```

174. Concatenate Characters

Given an input string array, write a program to get the second character of each string and form a new String by concatenating the fetched characters together. Print the new string formed.

Business Rules : 1. If the given input array element contains numbers, then print -1. 2. If the given input array element contains special characters , then print -2. 3. If the input array contains only one string, then print -3.

Create a class named UserProgramCode that has the following static method

public static string concatCharacter(string[] input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array .

The next 'n' lines of input correspond to elements in the input array.

Refer business rules and sample output for formatting specifications. Sample Input 1 :

3

ab

aaa

adbcd

Sample Output 1 :

bad Sample Input 2 :

4

ban

b%a

ssm

tea

Sample Output 2 :

-2

----Concatenate Characters---

```
using System;

using System.Collections.Generic;

using System.Linq;
```

```
using System.Text;


namespace concatcharacters
{
   class Program
   {
      static void Main(string[] args)
      {
         int i,n;
         n=Convert.ToInt32(Console.ReadLine());
         string[]s=new string[n];
         for (i = 0; i < n; i++)
         {
            s[i]=Console.ReadLine();
         }
         concat c = new concat();
         string output=c.concatenation(s);
         Console.WriteLine(output);
         Console.ReadLine();


      }
   }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace concatcharacters
{
   class concat
   {
      public string concatenation(string[] s)
      {
         StringBuilder sb = new StringBuilder();
         if (s.Length == 1)
         {
            return "-3";
         }
         foreach (string i in s)
         {
            foreach (char c in i)
            {
```

```
            if(Char.IsDigit(c))
            {
               return "-1";
            }
            if (!Char.IsLetter(c))
            {
               return "-2";
            }
          }
          sb.Append(i[1]);
        }
        return sb.ToString();

    }
  }
}
```

175.Validate ID Locations

Write a program to read two string inputs and check whether the first string is in valid format. First string is ID and second string is location. A valid ID should be in the format CTS-LLL-XXXX where LLL is the first three letters of given location and XXXX is a four digit number. If the given ID is as per the given format, print “valid” else print “invalid”. Example: Input1 = CTS-hyd-1234 Input2 = hyderabad output = valid

Include a class UserProgramCode with a static method validateIDLocations which accepts two Strings. The return value (Integer) should be 1 if the first string is valid, else return -1.

Create a Class Program which would be used to read 2 strings and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of 2 strings. Output consists of a string, “valid” or “invalid”.

Refer sample output for formatting specifications.

Sample Input 1:

CTS-hyd-1234 hyderabad Sample Output 1:

valid

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace idValidation
{
   class Program
   {
      static void Main(string[] args)
      {
         string s1 = Console.ReadLine();
         string s2 = Console.ReadLine();
        int i = UserProgramCode.validateIDlocations(s1, s2);
         if (i == 1)
         {
            Console.WriteLine("valid");
         }
         else
         {
            Console.WriteLine("Invalid");
         }
         Console.ReadLine();
```

```
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;

namespace idValidation
{
    class UserProgramCode
    {
        public static int validateIDlocations(string s1, string s2)
        {
            int output = 0;
            Regex reg = new Regex(@"^([CTS]+[-]+([A-Za-z]{3})+[-]+([0-9]{4}))$");
            if (reg.IsMatch(s1))
            {
                string res = s2.ToLower();
                if (s1.Contains(res.Substring(0, 3)))
                {
                    output = 1;
                }
                else
                {
                    output = -1;
                }
            }
```

```
        else
        {
           output = -2;
        }
        return output;
     }
  }
}
```

176. Next Year Day

Write a program to read a date String in dd/mm/yyyy format and to calculate the day which falls on the same date next year and print it. Note - return the output in small case.

Example:

Input = 13/07/2012

output = saturday

Include a class UserProgramCode with a static method nextYearDay which accepts a String. The return type (String) should return the day which falls on the same date next year. Return -1 in case the format of the date is incorrect.

Create a Class Program which would be used to accept an Integer, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a String, date in dd/mm/yyyy format.

Output consists of a String, the the day which falls on the same date next year.

Refer sample output for formatting specifications.

Sample Input:

13/07/2012

Sample Output:

saturday

72.Next year day

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication27
{
class UserProgramCode
  {
     public static string nextYearDay(string date)
     {
        string output = "";
        DateTime day;
        bool check = DateTime.TryParseExact(date, "dd/MM/yyyy", null,
System.Globalization.DateTimeStyles.None, out day);
        if (check == true)
        {
           day = day.AddYears(1);
           //Console.WriteLine(day);
           //day = day.DayOfWeek;
           output = day.DayOfWeek.ToString();
           output = output.ToLower();
           return output;
        }
        else
        {
           return output;
        }

     }
```

```
    class Program
    {
       static void Main(string[] args)
       {
          string input, output;
          input = Console.ReadLine();
          output = UserProgramCode.nextYearDay(input);
          if (output == null)
          {
             Console.WriteLine("Invalid date");
          }
          Console.WriteLine(output);
          Console.ReadKey();

       }
    }
  }
}
```

177. Count Vowels

Write a program to count the character which comes under the vowels sound from the given string .

The string value should have only the alphabet values.

Business Rules:

1. If the input string consists of any other character than the alphabets, return -1 from the method and print "Other characters found" in Main.

Include a class UserProgramCode with static method countVowels() that accepts a string and returns an integer.

Create a class Program which would get the input and call the static method countVowels() present in the UserProgramCode.

Input and Output Format: Input is a string. Output is an integer if the method returns the count, else a

String "Other characters found" if the method returns '-1'. Sample Input 1: sang-gee- Sample Output 1: Other characters found Sample Input 2: god Sample Output 2: 1

53.CountVowels

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace CountVowels
{
   class Program
   {
      static void Main(string[] args)
      {
         string str = Console.ReadLine();
         int rslt = UserProgramCode.countVowels(str);


         if (rslt == -1)
            Console.WriteLine("Other character found");
         else
            Console.WriteLine(rslt);
```

```
            Console.ReadLine();
        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;

namespace CountVowels
{
    class UserProgramCode
    {

        public static int countVowels(string str)
        {
            string str1=str.ToLower();
            int rs=0;
            Regex reg = new Regex("^[a-z]+$");
            if (reg.IsMatch(str1))
            {
                foreach (char c in str1)
```

```
            {
                if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u')
                {
                    rs++;
                }
            }
            // return rs;


        }
        else
        {
            rs = -1;
        }


        return rs;
    }
  }
}
```

Program 2:

12)/////count vowels///

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace CountVowells
{
   class UserProgramCode
   {
      public static int countVowels(string input)
      {
         int output = 0;
         foreach (char c in input)
         {
            if (char.IsLetter(c) == false)
            {
               output = -1;
               return output;
            }
            else
            {
               switch (c)
               {
```

```
                case 'a':
                case 'e':
                case 'i':
                case 'o':
                case 'u':
                case 'A':
                case 'E':
                case 'I':
                case 'O':
                case 'U':
                    output++;
                    break;

                }

            }

        }
        return output;

    }
  }
}
```

178.Repeated Integers

Write code to pick all the repeated integers in a given integer array, sort them in ascending order and put them in the output list. Print the output list.

Include a class UserProgramCode with a static method findRepeatedIntegers which accepts the size of an integer array and an integer array. The return type is void. Print the repeated integers in sorted order if present. If there are no repeated numbers, then print “No repeated numbers”. If there are negative numbers in the array, print “Array contains negative numbers” . Create a Class Program which would be used to accept Input array and call the static method present in UserProgramCode.

Input and Output Format:
Input consists of n+1 integers. The first integer corresponds to n, the number of elements in the array. The next 'n' integers correspond to the elements in the array.

Refer sample output for formatting specifications.

Assume that the maximum number of elements in the array is 30.

Sample Input 1:
4
3
3
2
10
Sample Output 1:
3

Sample Input 2:
4
3

1

2

10

Sample Output 2:

No repeated numbers

Sample Input 3:

4

3

-11

2

10

Sample Output 3:

Array contains negative numbers

98.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;


namespace ConsoleApplication98
{
   class Program
   {
      static void Main(string[] args)
      {
```

```
        int l=1;
        int[] a = new int[30];
        for (int i = 0; i <= l; i++)
        {
          a[i] = Convert.ToInt16(Console.ReadLine());
          l = a[0];
        }
        UserProgramCode.findRepeatedIntegers(a);
    }
}
class UserProgramCode
{
    public static void findRepeatedIntegers(int[] a)
    {
        ArrayList a1 = new ArrayList();
        int flag = 0;
        for (int i = 1; i <= a[0]; i++)
        {
            if (a[i] >= 0)
            {
                int c = 0;
                for (int j = 1; j <= a[0]; j++)
                {
                    if (a[i] == a[j])
                        c++;
                }
                if (c > 1)
                {


                    if (!(a1.Contains(a[i])))
                        a1.Add(a[i]);
```

```
            }
          }
          else
          {
            flag = 1;
            break;
          }
        }
        if (flag == 0)
        {
          a1.Sort();
          int c1 = 0;
          foreach (int i in a1)
            c1++;
          if (c1 == 0)
            Console.WriteLine("No repeated numbers");
          else
          {
            foreach (int i in a1)
            {
              Console.WriteLine(i);
            }
          }
        }
        else
          Console.WriteLine("Array contains negative numbers");
      }
    }
}
```

179. Number Validation

Write a program to read a string of 10 digit number and to check whether the string contains a 10 digit

number in the format XXX-XXX-XXXX where 'X' is a digit.

Include a class UserProgramCode with a static method validateNumber which accepts a string as input and returns an Integer .
The method returns 1 if the string meets the above specified format . If the string input does not meet the specified format the method returns -1.

Create a class Program which would get the input as a String and call the static method validateNumber present in the UserProgramCode.

Input and Output Format:
Input consists of a string.
Output is a string specifying the given string is valid ("Valid number format") or not ("Invalid number format") .
Refer sample output for formatting specifications.

Sample Input 1:
123-456-7895
Sample Output 1:
Valid number format
Sample Input 2:
-123-12344322
Sample Output 2:
Invalid number format

62

```
class Program
{
   static void Main(string[] args)
   {
      int f = 0;
```

```
        string s;
        s = Console.ReadLine();
        userprogramcode obj = new userprogramcode();
        f=obj.validatenumber(s);
        if(f==1)
            Console.WriteLine("Valid number format");
        if(f==-1)
            Console.WriteLine("Invalid number format");


    }
}
public class userprogramcode
{
    public int validatenumber(string s)
    {
        if (Regex.IsMatch(s, @"^\d{3}[-]\d{3}[-]\d{4}$"))
        {
            return 1;
        }
        else
            return -1;
    }
}
```

180. Validate String

For a given String apply the following validations. 1. The given input String should be only four characters long. If not print -1. 2. First character can be an alphabet or digit. If not print -2 . 3. Second character must be uppercase alphabet. (eg 'M','R'.. any alpbabet A - Z). If not print -3 . 4. Third character must be a number and also between 5-9. If not print -4 . If all the conditions are satisfied print 1. Example 1: input='vM7u3' output = -1 Example 2: input='&Mau' output = -2 Example 3: input='vrau' output = -3 Example 4: input='vR3a' output = -4 Example 5: input='vR5a' output = 1

Create a class named UserProgramCode that has the following static method

public static int validateString(string input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of a string.

Output is an integer.

Sample Input :

vR5a

Sample Output :

1

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace Validatestring

{
```

```
    class Program
    {
        static void Main(string[] args)
        {
            string str = Console.ReadLine();
            int op = UserProgramCode.validate(str);
            Console.WriteLine(op);
            Console.ReadLine();
        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Validatestring
{
    class UserProgramCode
    {
        public static int validate(string str)
        {
```

```
if (str.Length > 4)
{
    return -1;

}
```

```
else if (!Char.IsLetterOrDigit(str[0]))
{
    return -2;
}
```

```
else if (!(str[1] >= 65 && str[1] <= 90))
{
    return -3;
}
```

```
else if (!((str[2] == '5') || (str[2] == '6') || (str[2] == '7') || (str[2] == '8') || (str[2] == '9')))
{
```

```
            return -4;
        }

        else

            return 1;


    }

  }
}
```

181. IP Validator

Write code to read an IP address in a String variable and validate the IP address. Print “Valid” if it is a valid IP address else print “Invalid”.

Note: An IP address has the format a.b.c.d where a,b,c,d are numbers between 0-255

Include a class UserProgramCode with a static method ipValidator which accepts a string. The return type (integer) should return 1 if it a valid IP, else return 2.
Create a Class Program which would be used to accept a string and call the static method present in UserProgramCode.
Input and Output Format:
Input consists of a String.
Output consists of a String(“Valid” or “Invalid”).

Refer sample output for formatting specifications.

Sample Input 1:

132.145.184.210

Sample Output 1:

Valid

Sample Input 2:

132.145.184.290

Sample Output 2:

Invalid

Q3.IP Validator

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace level2
{
   class Program
   {
      static void Main(string[] args)
      {
                  string str = Console.ReadLine();
```

```
    int no = UserProgramCode.ipValidator(str);
        if(no==1)
            Console.WriteLine("Valid");
        else
            Console.WriteLine("Invalid");
    }
}
class UserProgramCode
{

    public static int ipValidator(string str)
    {

        Int32 i = 0, len, no,flag=0;
        len = str.Length;

        Int32 start = 0, count = 0;
        while (i < len)
        {
            if (str.ElementAt(i) != '.')
            {
                count++;
            }
```

```
    else
    {
        no = Int32.Parse(str.Substring(start, count));
            if (no < 0 || no > 255)
                return 2;
            start = start + count+1;

            flag++;
        count = 0;
    }
    i++;
}



no = Int32.Parse(str.Substring(start, count));
flag++;

if (no < 0 || no > 255)
    return 2;
else if (flag > 4)
    return 2;
else
{
```

```
            if (i == len)
            {
               return 1;
            }
            else
               return 0;
         }}

      }
   }
```

182. Fibonacci Series

Write method to generate fibonacci series and calculate the sum of first n numbers in the series and return it as output.

Example:

input = 5

output = 0 + 1 + 1 + 2 + 3 = 7

Include a class UserProgramCode with a static method getSumOfNfibos that accepts an integer as input and returns an integer.

Create a class Program which would get the input and call the static method getSumOfNfibos present in the UserProgramCode.

Input and Output Format: Input consists of an integer that corresponds to n. Output consists of an integer which corresponds to the sum of the first n terms in the fibonocci series.

Note: First two numbers in a Fibonacci series are 0, 1 and all other subsequent numbers are sum of its previous two numbers. Example - 0, 1, 1, 2, 3, 5...

Sample Input 1: 15 Sample Output 1: 986 Sample Input 2: 4 Sample Output 2: 4

n..27

```
class Program
  {
    static void Main(string[] args)
    {
      int n = Convert.ToInt16(Console.ReadLine());
      int i = UserProgramCode.getSumOfNfibos(n);
      Console.WriteLine(i);
      Console.ReadLine();
    }

  }
  class UserProgramCode
  {
    public static int getSumOfNfibos(int n)
    {
      int f = 0, f1 = -1, f2 = 1, sum = 0;
      for (int i = 0; i < n; i++)
```

```
        {
            f = f1 + f2;
            f1 = f2;
            f2 = f;
            sum = sum + f;
        }
        return sum;
    }
}
```

183. Rearrange Case

Given a string input, write a program to form a new string provided with the the below limitations 1. Check for the alphabet which has maximum number of Upper case and lower case in the input string value. 2. Uppercase alphabets would be moved to the start of the Output string and lowercase alphabets should be moved to the end of the string. 3. Remaining other alphabets will remain the same in between the start and end of the output variable irrespective of the case. Business rule: 1) If the Input string contains any special characters, then print 'Invalid Input'. 2) If the Input string does not contain Uppercase at all, then print 'Condition does not meet' . 3) If two or more alphabets has maximum upper and lower case, then print 're-arranging is not possible' . Create a class named UserProgramCode that has the following static method

public static string rearrangeCase(string input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:

Input consists of a string.

Output consists of a string.

Refer business rules and sample output for the format.

Sample Input 1 :

CancelPolicy Sample Output 1 :

CanelPoliycc Sample Input 2 :

XYZbossxyz Sample Output 2 :

Re-arranging is not possible

97.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;

namespace ConsoleApplication97
{
   class Program
   {
      static void Main(string[] args)
      {
         string a, b;
```

```
        a = Console.ReadLine();
        b=UserProgramCode.rearrangeCase(a);
        Console.WriteLine(b);



    }
}
class UserProgramCode
{
    static char cf;
    public static string rearrangeCase(string input1)
    {
        int l = input1.Length;
       // string a=Convert.ToString();
        int c1 = 0,c3=0;



        StringBuilder a1 = new StringBuilder();
        StringBuilder a2 = new StringBuilder();
        StringBuilder a3 = new StringBuilder();



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

```
{
   if(char.IsLower(input1[i]))
   {
      c1++;
   }
}
if (c1 == l)
{
   return ("Condition does not meet");
}
else
{



   for (int i = 0; i < l; i++)
   {
      int c2 = 0;
      if (char.IsUpper(input1[i]))
      {
         for (int j = 0; j < l; j++)
         {
            if (input1[j] == char.ToLower(input1[i]))
               c2++;
```

```
        }
    }
    if (c2 > c3)
    {
        c3 = c2;
        cf = input1[i];
    }
    else if (c2 == c3)
    {
        return ("Re-arranging is not possible");
    }

}
for (int i = 0; i < l; i++)
{
    if (input1[i] == char.ToUpper(cf))
    {
        a1.Append(cf);
    }
    else if (input1[i] == char.ToLower(cf))
    {
        a2.Append(char.ToLower(cf));
    }
```

```
            else
            {
               a3.Append(input1[i]);
            }
          }
        }
        return (a1.ToString() + "" + a3.ToString() + "" + a2.ToString());
      }
   }
}
```

184. Bill Amount Calculation

Raviraj group is having guest houses at different locations. Write a program to calculate bill amount as per the Room type & number of days of stay at a particular location. Rent for Non-AC room: Chennai(C): 1000 per day Hyderabad(H): 800 per day Bangalore(B): 1100 per day Rent for AC room: Chennai(C): 1300 per day Hyderabad(H): 1100 per day Bangalore(B): 1400 per day Input1: Location Input2: Room Type Input3 :Number of days The input values should exactly be the same as 'C', 'H',' B' for the given locations Chennai, Hyderabad and Banglore respectively and AC, NAC for the AC and non AC room type respectively. If input for location and room type is not given as per above criteria, print -1. Only positive number should be given for number of days. If not, print -1.

Create a class named UserProgramCode that has the following static method

public static int calculateBillAmount(char input1, string input2, int input3)

The first input parameter refers to the location, the second parameter refers to the room type and the third refers to the number of days of stay. Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format: The first input corresponds to the location, the second input corresponds to the room type and the third input corresponds to the number of days of stay. Output consists of an integer.

Sample Input: C AC 5

Sample Output :

6500

10.BILL AMOUNT

```
using System;


class Program
  {
     public static void Main( string[] args )
     {
            char input1;
            string input2;
            int input3;
            input1 = Convert.ToChar(Console.ReadLine());
            input2 = Console.ReadLine();
            input3 = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine(UserProgramCode.calculateBillAmount(input1,input2,input3));
  }
}
  using System;


class UserProgramCode
```

```
{
  public static int calculateBillAmount(char input1,string input2,int input3)
  {

     if(input3>0)
     {
     if(input1=='C' && input2.Equals("NAC"))
        return input3*1000;
     else if(input1=='C' && input2.Equals("AC"))
        return input3*1300;
     else if(input1=='H' && input2.Equals("NAC"))
        return input3*800;
     else if(input1=='H' && input2.Equals("AC"))
        return input3*1100;
     else if(input1=='B' && input2.Equals("NAC"))
        return input3*1100;
     else if(input1=='B' && input2.Equals("AC"))
        return input3*1400;
     else
        return -1;
     }
     else
        return -1;
```

```
    Console.ReadLine();


 }
}
```

185. Valid Negative Number

Write a program to read a negative number as a String variable and to validate the number. If the given string contains a valid negative number print corresponding positive number else print “Invalid number” .

Example:
input = "-94923"
output = "94923"

Include a class UserProgramCode with a static method validateNumber which accepts a String. The return type (String) should return the corresponding output. If the input string is not a valid negative number, the method returns "-1".
Create a Class Program which would be used to accept a String, and call the static method present in UserProgramCode.
Input and Output Format:
Input consists of a String(a negative number).
Output consists of a String(the corresponding output).
Refer sample output for formatting specifications.

Sample Input 1:
-94923
Sample Output 1:
94923

Sample Input 2:

-13O

Sample Output 2:

Invalid number

13.Valid Negative Number

```
using System;


        using System.Collections.Generic;

        using System.Linq;

        using System.Text;


        namespace ConsoleApplication9

        {

    public class UserProgramCode

    {

       public static string validateNumber(string str)

       {

          str.ToCharArray();

          int temp = 0;

          if (str[0] == '-')

          {

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

```
            if (str[i] >= 48 && str[i] <= 57)
               temp = 1;
            else
               temp = 0;
         if (temp == 1)
         {
            str = str.Substring(1, str.Length-1);
            return str.ToString();
         }
         else
            return "-1";
      }
      else
         return "-1";
   }
}
        class Program
        {
           static void Main(string[] args)
           {
      string str = Console.ReadLine();
      Console.WriteLine(UserProgramCode.validateNumber(str));
           }
```

```
    }
}
```

186. Interchange Characters

Write a program that accepts a string input and interchanges the first and last characters. Case sensitivity should be checked. Business rules: 1) Print 'Invalid String' when the given input string consists of any special characters or numbers. 2) Print 'No Change' when the first and last characters of the input string is same and of the same case. Example 1: Input: Execute Output: executE Example 2: Input: BoB Output: No Change Create a class named UserProgramCode that has the following static method

public static string interchangeFirstLast(string input1)

Create a class named Program that accepts the input and calls the static method present in the UserProgramCode.

Input and Output Format: Input consists of a string.

Output is a string. Refer sample output and business rules Sample Input 1: Execute

Sample Output 1:

executE

Sample Input 2: BoB

Sample Output 2:

No Change

84.)

```
using System;
using System.Collections.Generic;
```

```
using System.Linq;
using System.Text;



namespace ConsoleApplication13
{
   class userprogramcode
   {
      public static string getString(string ip1)
      {
         string[]  final = new string[ip1.Length];
         for (int j = 0; j < ip1.Length; j++)
            final[j] = ip1[j].ToString();
         string t1,ans="";
         for (int j = 0; j < ip1.Length; j++)
            if (!char.IsLetter(ip1[j]))
               return "Invalid String";
         if (final[0] != final[ip1.Length - 1])
         {
            t1 = final[0];
            final[0] = final[ip1.Length - 1];
            final[ip1.Length - 1] = t1;
            for (int i = 0; i < ip1.Length; i++)
```

```
            ans=ans+final[i];
        }
        else
          ans = "No Change";

        return ans;
    }
}
class Program
{
    static void Main(string[] args)
    {
        String x, y;

        x = Console.ReadLine();

        y = userprogramcode.getString(x);
        Console.WriteLine(y);

    }
```

```
    }
}
```

187. Common Characters

Write a method to count the common character from the given two Strings. Rule: - Space should not be counted as a letter. - Consider letters to be case sensitive. ie, 'a' is not equal to 'A'. Example: input1 = ""a black cow"" input2 =""battle ship"" output = 3 Include a class UserProgramCode with static method commonChars which accepts two String values.The return type is an interger. Create a class Program which would get the input and call the static method commonChars present in the UserProgramCode. Input Output format: The input consists of two strings. The output is an interger which counts the common characters.

Sample input 1: a black cow battle ship

Sample Output 1: 3

Sample input 2: australia sri lanka

Sample Output 2: 5

COMMON CHARACTER

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

class Program
{
```

```
    public static void Main()
    {
        String input1 = Console.ReadLine();
        String input2 = Console.ReadLine();
        Console.WriteLine(UserProgramCode.commonChars(input1, input2));
        Console.ReadLine();
    }
}

class UserProgramCode
{
    public static int commonChars(String input1, String input2)
    {
        char[] s1a = input1.ToCharArray();

        char[] s2a = input2.ToCharArray();

        int l1 = input1.Length;
        int l2 = input2.Length;

        Array.Sort(s1a);
        Array.Sort(s2a);
```

```
List<char> s1c = new List<char>();

List<char> s2c = new List<char>();


for (int i = 0; i < l1; i++)

{

   if (s1a[i] == ' ')

      continue;


   else if (!s1c.Contains(s1a[i]))

   {

      s1c.Add(s1a[i]);

   }

}


for (int i = 0; i < l2; i++)

{

   if (s2a[i] == ' ')

      continue;


   else if (!s2c.Contains(s2a[i]))

   {

      s2c.Add(s2a[i]);

   }
```

```
        }


        int c = 0;


        foreach (var i1 in s1c)
        {
            foreach (var i2 in s2c)
            {
                if (i1 == i2)
                {
                    c++;
                }
            }
        }


        // Console.WriteLine(c);
        return c;



    }
}
```

188. Repeated Words

Given two string inputs input1 and input2, write a program to calculate the number of times each word in input1 occurs in input2 and print the words which has occurred the maximum number of times in the output with empty spaces between them and also in the same order as given in input1. Output should be printed in lower case.Ignore case sensitiveness in both the input strings. Business Rules: 1)If any of the words is repetitive in input1, then print -1. 2)If none of the words from input1 has occurred in input2, then print -2. Create a class named UserProgramCode that has the following static method

public static string repeatedWords(string input1, string input2)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format: Input consists of 2 strings, input1 and input 2.

Output consists of a string.

Refer business rules and sample output for formatting specifications.

Sample Input 1 : Apple is a fruit apple is a nice fruit and apple is available in all seasons

Sample Output 1 :

apple is Sample Input 2 : Does he have to have a car ? Yes he should.

Sample Output 2 :

-1

REPEATED WORDS

```
using System;

  class Program

  {

     public static void Main( string[] args )

     {

                    string input1,input2,output;

                    input1 = Console.ReadLine();
```

```
                    input2 = Console.ReadLine();


                    output = UserProgramCode.repeatedWords(input1,input2);
                    Console.WriteLine(output);
      Console.ReadKey();
    }
  }
using System;
using System.Collections;


class UserProgramCode {
          public static string repeatedWords(string input1,string input2)
           {
      input1 = input1 + " ";
      input2 = input2 + " ";
      ArrayList s1 = new ArrayList();
      ArrayList s2 = new ArrayList();
      ArrayList intS1 = new ArrayList();
      string[] a1 = input1.Split(' ');
      string[] a2 = input2.Split(' ');
      int i,c=0,j,max=0,flag=0;
      string str="";
      s1.Add(a1[0]);
```

```
for (i = 1; i < a1.Length-1; i++)
{
   if (s1.Contains(a1[i]))
      return "-1";
   else
      s1.Add(a1[i]);
}
for (i = 0; i < a2.Length-1; i++)
   s2.Add(a2[i]);
for (i = 0; i < s1.Count; i++)
{
   for (c=0,j = 0; j < s2.Count; j++)
   {
      if (a1[i].Equals((string)s2[j],StringComparison.OrdinalIgnoreCase))
      {
         c++;
      }
   }
  intS1.Add(c);
}
for (i = 0; i < intS1.Count; i++)
{
   if ((int)intS1[i] == 0)
```

```
        flag++;
    }
    if (flag == intS1.Count)
       return "-2";
    max = (int)intS1[0];
    for (i = 0; i < intS1.Count; i++)
    {
       if ((int)intS1[i] > max)
       {
          max = (int)intS1[i];
       }
    }
    for (i = 0; i < s1.Count; i++)
    {
       if ((int)intS1[i] == max)
       {
          str = str + a1[i] + " ";
       }

    }
       //Console.ReadKey();
    return str.ToLower();
          }
```

}

189. Get All Elements

Write a program to get all the elements that are greater than 5 from a given input integer list. Display it in the order as present in the array.

Print the elements.

Example:

Input1: [1,3,7,8,5,13]

Output1:[7,8,13]

Business Rule:

If any of the element in the input list is greater than 500 then store -1 in the oth index of the output list.

Include a class UserProgramCode with a static method GetAllElements which accepts an integer List and its size. The return type (integer list) should return output according to the business ruless

Create a Class Program which would be used to accept a list, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of n+1 integers, where first integer corresponds to the size of the list, followed by the corresponding list elements.

Output consists of an Integer list, or a String “Array element greater than 500” if any of the elements is greater than 500.

Refer sample output for formatting specifications.

Sample Input 1:

6

1

3

7

8

5

13

Sample Output1:

7

8

13

Sample Input 2:

6

1

3

7

8

501

13

Sample Output 2:

Array element greater than 500

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace get_all_the_elements

{

    class Program
```

```
{
   static void Main(string[] args)
   {
      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[] op = UserProgramCode.getElements(arr);
      int len=op.Length;
      for (int i = 0; i < op[len - 1]; i++)
      {
         if (op[0] == -1)
         {
            Console.WriteLine("Array element greater than 500");
         }
         else
            Console.WriteLine(op[i]);
      }
      Console.ReadLine();

   }
```

```
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace get_all_the_elements
{
    class UserProgramCode
    {
        public static int[] getElements(int[] arr)
        {
            int k=0;
            int[] temp =new int[30];
            for (int i = 0; i < arr.Length; i++)
            {
                if (arr[i] > 500)
                    temp[0] = -1;
            }
```

```
            for (int i = 0; i < arr.Length; i++)
            {
               if (arr[i] > 5)
               {
                  temp[k] = arr[i];
                  k++;
               }

            }
            temp[29] = k ;
         return temp;
      }
   }
}
```

190. Calculate Telephone Bill

Write a program which reads the number of calls as input and calculates the monthly telephone bills as per the following rules. Print the bill amount.

Minimum Rs. 200 for upto 300 calls.
Plus Rs. 0.60 per call for next 50 calls.
Plus Rs. 0.50 per call for next 50 calls.
Plus Rs. 0.40 per call for any call beyond 400 calls.

Example :

If the calls is 720, the calculation would be as follows :

1. First 300 calls the charge is Rs 200.
2. Next 50 the charge is 50 *.60=30.
3. Next 50 the charge is 50 *.50=25.
4. Balance calls (720-300-50-50) 320 the charge is 320 *.40=128.
5. Total charge = 200 + 30+25+128 = 383.

The calculated charge should be in double datatype which is rounded to 2 decimal places.

Include a class UserProgramCode with a static method calculateTelephoneBill which accepts an Integer. The return type (Double) should return the final bill amount.

Create a Class Program which would be used to accept an Integer, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of an Integer, which corresponds to number of calls.

Output consists of a Double (The final bill amount).

Refer sample output for formatting specifications.

Sample Input 1:

720

Sample Output 2:

383.00

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace calculate_telephone_bill
{
```

```
class UserProgramCode
{
    public static double Telephone(int input1)
    {
        double output1;
        if (input1 <= 300)
        {
            output1 = 200;
            return output1;
        }
        else if (input1 > 300 && input1 <= 350)
        {
            output1 = 200 + ((input1 - 300) * 0.60);
            return output1;
        }
        else if (input1 > 350 && input1 <= 400)
        {
            output1 = 200 + (50 * 0.60) + ((input1 - 350) * 0.50);
            return output1;
        }
        else if (input1 > 400)
        {
            output1 = 200 + (50 * 0.60) + (50 * 0.50) + ((input1 - 400) * 0.40);
            return output1;
        }
        return 0;

    }
}
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace calculate_telephone_bill
{
   class Program
   {
      static void Main(string[] args)
      {
         int Calls = int.Parse(Console.ReadLine());
         double op = UserProgramCode.Telephone(Calls);
         Console.WriteLine(op.ToString("F"));
         Console.ReadLine();

      }
   }
}
```

191. Calculate Take Home Salary

TMB Software solution is under developing software company in KolKatta. The company is providing the Provident fund for all the employees based on the salary criteria as given below. Write a method to find the take home salary for the employee. Medical Insurance for each employee is Rs 678. Salary Range---------- PF Lesser than 15000 ------- 750 15001 - 22000 -------- 850 22001-30000 -------- 925 Above 30000 -------- 1000 Take Home Salary = Salary - pf - MedicalInsurance Include a class UserProgramCode with a static method calculateHomeSalary which accepts an integer and returns an integer that corresponds to the Take Home Salary.

The method returns -1 when the input integer is negative. Create a class Program which would get the input and call the static method calculateHomeSalary present in the UserProgramCode. If the method returns -1, print 'Invalid Input'.

Input and Output format : Input consists of a integer that represents a salary. Output is an integer that corresponds to 'take home salary' or a string 'Invalid Input'.

Sample Input 1 : 13500

Sample Output 1 : 12072

Sample Input 2 : -10000

Sample Output 2 : Invalid Input

20.CALCULATE TAKE HOME SALARY

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace takehomesalary
{
   class Program
   {
      static void Main(string[] args)
      {
         int rslt;
         Console.WriteLine("enter the salary");
         int Salary= Convert.ToInt32(Console.ReadLine());
         rslt=UserProgramCode.calculateHomeSalary(Salary);
         if (rslt == -1)
```

```
        {
            Console.WriteLine("Invalid Input");
        }
        else
        {
            Console.WriteLine(rslt);
        }
        Console.ReadLine();
      }
   }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace takehomesalary
{
   class UserProgramCode
   {
      public static int calculateHomeSalary(int Salary)
      {
```

```
int PF=0;
int MedicalInsurance=678;

int HomeSal=0;
if (Salary > 0)
{
   if (Salary < 15000)
   {
      PF = 750;
      HomeSal = (Salary - PF - MedicalInsurance);

   }
   else if (Salary >= 15001 && Salary <= 22000)
   {
      PF = 850;
      HomeSal = Salary - PF - MedicalInsurance;

   }
   else if (Salary >= 22001 && Salary <= 30000)
   {
      PF = 925;
      HomeSal = Salary - PF - MedicalInsurance;
```

```
            }
            else if (Salary > 30000)
            {
               PF = 1000;
               HomeSal = Salary - PF - MedicalInsurance;

            }
            return HomeSal;
         }
         else
         {
            return -1;
         }

      }
   }
}
```

192. Count the number of odd integers

Write a code to count the number of odd integers in the given integer array. Include a class UserProgramCode with static method countOddIntegers that accepts an integer array and the return type should be int (count of odd Integers). Return -1 if the array contains negative values. Create a class Program which would get the input and call the static method countOddIntegers present in the

UserProgramCode. In Program display “The Array consists non-positive value(s)” if -1 is returned, else print the count. Input and Output Format:

Input consists of n+1 integers. The first integer corresponds to n, the number of elements in the array. The next 'n' integers correspond to the elements in the array.

Output consists of an integer or a string. Refer sample output for formatting specification. SAMPLE INPUT 1: 2 -1 2 SAMPLE OUTPUT 1: The Array consists non-positive value(s) SAMPLE INPUT 2: 2 1 3 SAMPLE OUTPUT 2: 2

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace count_odd_integers
{
   class Program
   {
      static void Main(string[] args)
      {
         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 op = UserProgramCode.countOdd(arr,n);
            if (op == -1)
            {
                Console.WriteLine("The Array consists non-positive value(s)");

            }
            else
                Console.WriteLine(op);
            Console.ReadLine();
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace count_odd_integers
{
    class UserProgramCode
```

```
  {
    public static int countOdd(int[] arr,int n)
    {
      int c = 0;
      for (int i = 0; i < n;i++ )
      {
        if (arr[i] < 0)
          return -1;
      }

        for (int i = 0; i < n; i++)
        {
          if (arr[i] % 2 != 0)
          {
            c++;
          }
        }
      return c;
    }
  }
}
```

193.Calculate New Salary

The HR of an IT company fixes the salary of a new joinee who joins the company with the previous experience. Given the year of experience, Technology expertise and the previous salary drawn, the HR fixes the current salary as follows: 1. Straight 30% hike from the previous salary drawn. 2. a. Add Extra 5% if the year of experience is more than 3 and less than or equal to 5 years. b. Add Extra 10% if the year of experience is more than 5 and less than or equal to 8 years. c. Add Extra 15% if the year of experience is more than 8 years 3. Technology expertise is classified broadly into two namely common skills (CS) and rare skills (RS) People having rare skills get an extra 5% increase from their previous salary drawn. Write a program to calculate the salary the HR fixes for the new joinee given the experience (input1), technology expertise classification (input2) and the previous salary drawn (input3) and print the output in the given format

Your Salary is fixed as Rs YYYYYY where YYYYYY is the calculated new salary. (The digits depends upon the salary calculated.) Salary is rounded off and displayed as an integer.

Business rules: 1. If the experience is given more than 25 or less than 0, then print “Invalid Experience” 2. If the technology expertise classification is given other than CS or RS, then print “Invalid Technology expertise classification” 3. If the previous salary is given more than 100000 or less than 0, then print “Invalid Salary”

Create a class named UserProgramCode that has the following static method

public static int calculateNewSalary(int,string,int)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer that corresponds to experience (input 1).

The second line of the input consists of a string that corresponds to technology expertise (input 2).

The third line of the input consists of an integer that corresponds to previous salary drawn (input 3).

Refer business rules and sample output for formatting specifications. Sample Input 1 : 7 RS 30000

Sample Output 1 :

Your Salary is fixed as Rs 43500 Sample Input 2 : 7 RSS 30000

Sample Output 2 :

Invalid Technology expertise classification

```
class Program
{
   public static void Main(string[] args)
   {
      int experience, oldSalary, salary;
      string expertise;
      experience = Convert.ToInt32(Console.ReadLine());
      expertise = Console.ReadLine();
      oldSalary = Convert.ToInt32(Console.ReadLine());
      salary = UserProgramCode.calculateNewSalary(experience, expertise, oldSalary);
      if (salary == -1)
         Console.WriteLine("Invalid Experience");
      else if (salary == -2)
         Console.WriteLine("Invalid Technology expertise classification");
      else if (salary == -3)
         Console.WriteLine("Invalid Salary");
      else
         Console.WriteLine("Your Salary is fixed as Rs {0}", salary);
      Console.ReadLine();
   }
}

class UserProgramCode
```

```
{
  public static int calculateNewSalary(int experience,string expertise,int oldSalary)
  {
     double sal = 0,ctr=0,sa=0;
     if (experience > 25 || experience < 0)
     {
        sa = -1;
        ctr++;



     }
     if (expertise != "RS" && expertise != "CS")
     {
        sa = -2;
        ctr++;


     }
     if (oldSalary > 100000 || oldSalary < 0)
     {
        sa = -3;
        ctr++;


     }
```

```
if(ctr==0)
 {
    sal = sal + (oldSalary * 0.3);
    if (experience > 3 && experience <= 5)
    {
       sal = sal + (oldSalary * 0.05);
    }
    else if (experience > 5 && experience <= 8)
    {
       sal = sal + (oldSalary * 0.10);
    }
    else if (experience > 8)
    {
       sal = sal + (oldSalary * 0.15);
    }
    if(expertise=="RS")
    {
       sal=sal+(oldSalary*0.05);
    }
    sal = sal + oldSalary;
    sa = Math.Round(sal);

 }
```

```
    return (int)sa;
  }
}
```

194. Count Sequential Characters

Write a program to count the number of characters which gets repeated 3 times consecutively. If no character gets repeated 3 times consecutively, then print -1. Create a class named UserProgramCode that has the following static method

public static int countSequentialChars(string input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:

Input consists of a string.

Output is an integer.

Sample Input 1 : abcXXXabc

Sample Output 1 :

1 Sample Input 2 :

aaaxxyzAAAx

Sample Output 2 :

2

66)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication24
{
   class Program
```

```
{
   static void Main(string[] args)
   {
      string s = Console.ReadLine();
      int a = UserProgramCode.countSequentialChars(s);
      if(a==-1)
         Console.WriteLine("No Repeated Words Found");
      else
         Console.WriteLine(a);
      Console.ReadLine();

   }
}
class UserProgramCode
{
   public static int countSequentialChars(string s)
   {

      int l = s.Length;
      string[] st = new string[50];

      for (int k = 0; k < l; k++)
      {
         st[k] = s.Substring(k, 1);



      }

      int count = 0;
      int c=0;

      for (int k = 0; k < l - 1; k++)
```

```
        {

            if (st[k] == st[k+1])
               count++;
            else
               count = 0;
            if (count == 2)
               c++;


          }


        if(c==0)
        return -1;
        else
           return c;




     }
  }
}
```

195. Find Total number of days in given month

Write code to find out total number of days in the given month for the given year.

Month is coded as: Jan=0, Feb=1 ,Mar=2 ...

Include a class UserProgramCode with static method getNumberOfDays that accepts two integers and return type should be int.

Create a class Program which would get the input and call the static method getNumberOfDays(int

year, int month) present in the UserProgramCode. Return the result from getNumberOfDays and dispaly the result in Program class.

Input and Output Format : The first integer represent the year. The second integer represents the month The output is an interger which is number of days in the given month. SAMPLE INPUT 1: 2000 1 SAMPLE OUTPUT 1: 29

44. Find total number of days in given month

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Workout44
{
   class UserProgramCode
   {
      public static int getNumberOfDays(int y,int m)
      {
         int d;
         d = System.DateTime.DaysInMonth(y,m+1);
         return d;
      }
   }

   class Program
   {
      static void Main(string[] args)
      {
         //UserProgramCode u = new UserProgramCode();
        int y,m,n;
        y = int.Parse(Console.ReadLine());
         m= int.Parse(Console.ReadLine());
```

```
        n = UserProgramCode.getNumberOfDays(y,m);
        Console.WriteLine(n);

      }
   }
}
```

196. ValidateNumber

Write a code to check the if the given number is validate or not. This string is expected to contain a 10 digit number in the format XXX-XXX-XXXX where 'X' is a digit. Include a class UserProgramCode with static method validateNumber which accepts String value. The validateNumber function return type is an interger.Return 1 if the given string meets this format else return -1.if the function returns 1 then print as "Valid Number" and if it returns -1 then print as "Invalid Number".

Create a class Program which would get the input and call the static method validateNumber present in the UserProgramCode.

Input Output Format: The input consists of String. The output consists of a String "Valid Number" or "Invalid Number".

Sample input 1: 452-789-4568

Sample output 1: Valid Number

Sample input 2: 1234-234-123

Sample output 2: Invalid Number

62

```
class Program
{
   static void Main(string[] args)
   {
```

```
        int f = 0;
        string s;
        s = Console.ReadLine();
        userprogramcode obj = new userprogramcode();
        f=obj.validatenumber(s);
        if(f==1)
            Console.WriteLine("Valid number format");
        if(f==-1)
            Console.WriteLine("Invalid number format");

    }
}
public class userprogramcode
{
    public int validatenumber(string s)
    {
        if (Regex.IsMatch(s, @"^\d{3}[-]\d{3}[-]\d{4}$"))
        {
            return 1;
        }
        else
            return -1;
    }
```

}

197. Print Digit Sum

Write a program that accepts a string input and finds the sum of all numeric digits present in the string.

Example 1: input : abc12de4 output : 7 Example 2: input : udjc&23er output : -1 Business Rules : 1. If the given input string contains any special characters, then print -1. 2. f the given input string contains no numbers,then print -2. Create a class named UserProgramCode that has the following static method

public static int getdigits(string input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format: Input consists of a String

Output is an integer.

Sample Input 1:

abc12de4

Sample Output 1:

7

Sample Input 2:

udjc&23er

Sample Output 2:

-1

Program 75:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;
```

```
using System.Text.RegularExpressions;


namespace ConsoleApplication18
{
   class Program
   {
      static void Main(string[] args)
      {
         string s=Console.ReadLine();
         int i=userProgramCode.getdigits(s);
         Console.WriteLine(i);
         Console.ReadLine();
      }
   }
   class userProgramCode
   {
      public static int getdigits(string i)
      {
         int s=0;
         for (int j = 0; j < i.Length; j++)
         {
            if ((char.IsLetterOrDigit(i[j])))
            {
```

```
            if (char.IsDigit(i[j]))
                s += (Convert.ToInt32(i[j])-48);
            }
            else
                return -1;
        }
        if (s == 0)
            return -2;
        else
            return s;
    }
  }
}
```

198. Strong Number

Write a program to find whether the given integer input is a strong number or not. If the sum of each digits factorial is the same as the given input value then it is a strong number. If the Input1 is strong number then print "Input1 is a Strong Number" where Input1 is the input integer value. (Refer Example) Business rule: 1) If the Input1 value is not a strong number then print "Sum of all digits factorial is XX" where XX is the total of each digits factorial value. 2) Print "Invalid Input" when given input number is a negative number. Example:1 Input1: 145 1!+4!+5! = 1+24+120 = 145 Output1: 145 is a Strong Number Example:2 Input1: 25 2!+5! = 2+120 = 122 Output1: Sum of all digits factorial is 122

Create a class named UserProgramCode that has the following static method

public static String checkStrongNumber(int input1)

Create a class named Program that accepts the inputs and calls the static method present in the

UserProgramCode.

Input and Output Format: Input consists of a single integer.

Output is a string.

Sample Input 1:

145

Sample Output 1:

145 is a Strong Number

Sample Input 2:

25

Sample Output 2:

Sum of all digits factorial is 122

Program 74:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;

namespace ConsoleApplication18
{
   class Program
   {
```

```
    static void Main(string[] args)
    {
        int n = Convert.ToInt32(Console.ReadLine());
        string r = userProgramCode.checkStrongNumber(n);
        Console.WriteLine(r);
        Console.ReadLine();
    }
}
class userProgramCode
{
    public static String checkStrongNumber(int i)
    {
        if (i < 0)
            return "Invalid Input";
        int t, r, f = 0;
        t = i;
        while (t > 0)
        {
            int fact = 1;
            r = t % 10;
            for (int j = 1; j <= r; j++)
            {
                fact *= j;
```

```
            }
            f += fact;
            t /= 10;
        }
        if (f == i)
            return f + " is a Strong Number";
        else
            return "Sum of all digits factorial is " + f;

    }
  }
}
```

199.Find Span

Given an integer array as input, write a program to find the size of the largest Span in the given array,

Note: Span is the number of elements between two repeated numbers including both numbers.

Assume an array with single element has a span of 1. Business rule: If there is no number repeated in an array, print -1. If there are two repeated integers in the input array, consider the first number and return the span.

Create a class named UserProgramCode that has the following static method

public static int getMaxSpan(int[] input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

The first line of the input consists of an integer, n that corresponds to the number of elements in the input array .

The next 'n' lines of input correspond to elements in the input array.

Refer business rules and sample output for formatting specifications. Sample Input 1 : 5 1 2 1 1 3

Sample Output 1:

4 Sample Input 2 : 7 1 4 2 1 4 1 5

Sample Output 2 :

6

```
 using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace span

{

   class UserProgramCode

   {

      public static int getMaxSpan(int size, int[] arr)

      {

         //Fill your code here

         int i = 0; int j = 0;

         int span = 1; int temp = 0;

         int flag = 0;

         for (i = 0; i < size; i++)

         {

            for (j = i + 1; j < size; j++)
```

```
    {
        if (arr[i] == arr[j])
        {
            flag = 1;
        }
    }
}
if (flag == 0)
    return -1;

for (i = 0; i < size; i++)
{
    for (j = i + 1; j < size; j++)
    {
        if (arr[i] == arr[j])
        {
            temp = j - i + 1;
            if (temp > span)
            {
                span = temp;
            }
        }
    }
```

```
            }


            return span;
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace TestPractice
{
    class Program
    {
        static void Main(string[] args)
        {
            int size;
            size = Convert.ToInt32(Console.ReadLine());
            int[] arr = new int[size];
```

```
        for (int i = 0; i < size; i++)
        {
           arr[i] = Convert.ToInt32(Console.ReadLine());
        }
        int res = UserMainCode.getMaxSpan(size,arr);
        Console.WriteLine(res);
        Console.ReadLine();
      }
   }
}
```

200. Longest Palindrome

Given an input string input1, write a program to find the length of the longest substring which is a palindrome. Palindrome is a word, phrase, or sequence that reads the same backwards as forwards e.g. madam Ignore case sensitivity for the input strings. Business Rule: 1) If the input string contains any number,then print -1. 2) If the input string contains any special characters,then print -2. 3) If the input string does not contain a string palindrome,then print -3. Please note that a single character is not considered to be palindrome. Create a class named UserProgramCode that has the following static method

public static int longestPalindrome(string input1)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:

Input consists of a string.

Output is an integer.

Refer business rules and sample output for output format.

Sample Input 1: seaesstringnirts

Sample Output 1: 11 Sample Input 2: sea34esstringnirts Sample Output 2 :

-1

91.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Collections;
namespace ConsoleApplication23
{
   class UserProgramCode
   {
      public static int longestPalindrome(string s1)
      {
         string s = s1;
         s = Regex.Replace(s, @"\s+", " ");
         string[] s2 = s.Split(' ');
         char[] a2=new char[100];
```

```
for(int i=0;i<s2.Length;i++)
{
   for(int j=0;j<s2[i].Length;j++)
   {
      if(char.IsDigit(s2[i][j]))
      {
         return -1;
      }


   }
}
for (int i = 0; i < s2.Length; i++)
{
   for (int j = 0; j < s2[i].Length; j++)
   {
      if (!char.IsLetter(s2[i][j]))
      {
         return -2;
      }


   }
```

```
}



for (int i = 0; i < s2.Length; i++)
{
   int flag = 0;
   string r=s2[i];
   char[] a1=r.ToCharArray();
   int k = a1.Length;
   for (int i1 = 0; i < k; i1++)
   {

      a2[i1] = a1[k-1];
      k--;
   }
   for (int j = 0; j < s2[i].Length; j++)
   {



      if (s2[i][j] == a2[j])
      {
        flag = 1;
      }
```

```
            else
            {
                break;
            }


        }
        if (flag == 1)
        {
          return flag;
        }


    }


    return 0;

}


static void Main(string[] args)
{
```

```
        string a;
        a=Console.ReadLine();
        int flag;
        flag =longestPalindrome(a);
        Console.WriteLine(flag);
      }
   }
}
```

201. Find Occurence

Write a method to find the occurence (number of times) of a given character in a given input string. Include a class UserProgramCode with a static method findOccurence which accepts a string and character as input and returns an integer. Business Rules: 1. Search criteria is irrespective of cases. 2. The input string should consists of only alphabets, no special characters or numbers should be there in the string. If present, the method returns -1.. Create a class Program which would get the input and call the static method findOccurence present in the UserProgramCode. If the method returns -1, print 'Invalid Input'. Input and Output format : Input consists of string and character. Refer sample output for formatting specifications. Sample Input 1 : HELLO friends Welcome to CSharp wonderful world L Sample Output 1 : 5 Sample Input 2 : Gr8...I am fine. 8 Sample Output 2 : Invalid Input

69)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
```

```
{
    class Program
    {
        static void Main(string[] args)
        {

            string s = Console.ReadLine();
            char c = Convert.ToChar(Console.ReadLine());

            int a = UserProgramCode.findOccurence(s,c);
            if (a == -1)
            Console.WriteLine("Invalid Input");
            else
            Console.WriteLine(a);
            Console.ReadLine();



        }
    }
    class UserProgramCode
    {
        public static int findOccurence(string a,char b)
        {
```

```
        int count = 0;



        if (a.Any(ch => !(Char.IsLetter(ch)||Char.IsWhiteSpace(ch))))
           return -1;

        foreach (char e in a)
        {
          if (e==char.ToUpper(b)||e==char.ToLower(b))
              count++;
        }

        return count;









      }
   }
}
```