Q1.Triplets

solution:

program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Triplets

{

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

int[] r = Triplets1.findTriplets(arr, k);

foreach (int y in r)

{

Console.WriteLine(y);

}

Console.ReadKey();

}

}

}

Triplets.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Triplets

{

class Triplets1

{

public static int[] findTriplets(int[] input1, int z)

{

int num = input1.Length;

int l;

int[] b = new int[3];

l = 0;

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

{

if (input1[i] < 0)

{

b[0] = -1;

return b;

}

}

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

{

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

{

for (int k = j+1; k < num; k++)

{

if (input1[i] != input1[j] && input1[j] != input1[k] && input1[k] != input1[i])

{

if (input1[i] + input1[j] + input1[k] == z)

{

b[l] = input1[i];

b[l + 1] = input1[j];

b[l + 2] = input1[k];

l = l + 3;

}

}

else if (input1[i] == input1[j])

{

b[0] = -3;

return b;

}

else

{

b[0] = -2;

return b;

}

}

}

}

return b;

}

}

}

Q2. Strong Number.

Solution:

program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace StrongNUmber

{

class Program

{

static void Main(string[] args)

{

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

string r = userProgramCode.checkStrongNumber(n);

Console.WriteLine(r);

Console.ReadKey();

}

}

}

userProgramCode.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace StrongNUmber

{

class userProgramCode

{

public static string checkStrongNumber(int i)

{

if (i < 0)

{

return ("Invalid Input");

}

int t = i;

int f=0;

while (t > 0)

{

int fact = 1;

int rem = t % 10;

for (int j = 1; j <= rem; j++)

{

fact = fact \* j;

}

f = f + fact;

t = t / 10;

}

if (f == i)

{

return (f + " is a strong Number");

}

else

return ("Sum of all digits factorial is " + f);

}

}

}

Q3. Test Vowels Order

Solution:

program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace TestVowelsOrder

{

class Program

{

static void Main(string[] args)

{

string str1 = Console.ReadLine();

int r = usermainCode.testOrderVowels(str1);

Console.WriteLine(r);

Console.ReadKey();

}

}

}

userMainCode.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace TestVowelsOrder

{

class usermainCode

{

public static int testOrderVowels(string str)

{

int count = 0;

StringBuilder sb = new StringBuilder();

char[] ch = str.ToCharArray();

foreach (char i in ch)

{

if (char.IsLetter(i))

{

if (i == 'a' || i == 'e' || i == 'i' || i == 'o' || i == 'u')

{

sb.Append(i);

count++;

}

}

}

string st = sb.ToString();

char[] ch1=st.ToCharArray();

string st1= new string(ch1.Distinct().ToArray());

if (count == 5)

{

if (st == "aeiou")

return (1);

else

{

if(st1.Length<5)

return(-1);

else

return(2);

}

}

else

{

if(count>5)

return (-1);

}

return (0);

}

}

}

Q1) Count the occurrences of 3 consecutive repeated characters in the given input.

Sample Input:

abcXXXabc

Sample Output:

1

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Consecutive

{

class Program

{

static void Main(string[] args)

{

string ip = Console.ReadLine();

int res=maincode.HasConsecutiveChars(ip);

if (res!=0)

{

Console.WriteLine(res);

}

else

{

Console.WriteLine("No repeated chars");

}

}

}

}

UserMainCode.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Consecutive

{

class maincode

{

public static int HasConsecutiveChars(string source)

{

int count = 0;

if (string.IsNullOrEmpty(source))

{

return 0;

}

if (source.Length == 1)

{

return 0;

}

int charCount = 1;

for (int i = 0; i < source.Length - 1; i++)

{

char c = source[i];

if (c == source[i + 1])

{

charCount++;

if (charCount == 3)

{

count++;

}

}

else

{

charCount = 1;

}

}

return count;

}

}

}

Q2) All Vowels

Sample Input:

Alphabet

Invalid

Facetiously

Valid

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AllVowels

{

class Program

{

static void Main(string[] args)

{

string input = Console.ReadLine();

string result = UserMainCode.checkVowels(input);

Console.WriteLine(result);

}

}

}

UserMainCode.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AllVowels

{

class UserMainCode

{

public static string checkVowels(string input)

{

StringBuilder sb = new StringBuilder();

string s2 = "aeiou";

string result = string.Empty;

char[] arr = input.ToLower().ToCharArray();

foreach (var a in arr)

{

if (a == 'a' || a == 'e' || a == 'i' || a == 'o' || a == 'u')

{

sb.Append(a);

}

}

if (sb.ToString() == s2)

{

result="Valid";

}

else

{

result="Invalid";

}

return result;

}

}

}

Q3) Add Years

Sample Input:

12/07/1990(dd/MM/yyyy)

5

07/12/1995(MM/dd/yyyy)

UserMainCode.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AddYears

{

class AddYears

{

public static string addDate(string date,int years)

{

DateTime dt1;

string format="dd/MM/yyyy";

bool b = DateTime.TryParseExact(date, format, null, System.Globalization.DateTimeStyles.None, out dt1);

if (!b)

{

return "Invalid Input";

}

else

{

dt1 = dt1.AddYears(years);

return dt1.ToString("MM/dd/yyyy");

}

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AddYears

{

class Program

{

static void Main(string[] args)

{

string date = Console.ReadLine();

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

string result = AddYears.addDate(date, yr);

Console.WriteLine(result);

}

}

}

Q4) GCD Array

UserMainCode.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace GCDArray

{

class UserMainCode

{

public static int gcdArray(int[] arr, int n)

{

int count = 0;

if (n <= 0)

{

return -1;

}

else

{

int m = arr.Min();

while (m > 0)

{

count = 0;

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

{

if (arr[i] % m == 0)

{

count++;

}

}

if (count == n)

{

return m;

}

else

{

m--;

}

}

return m;

}

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace GCDArray

{

class Program

{

static void Main(string[] args)

{

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

int[] arr = new int[size];

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

{

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

}

int res = UserMainCode.gcdArray(arr, size);

Console.WriteLine(res);

}

}

}

Q5) OddSum

UserMainCode.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace OddEvenSum

{

class UserMainCode

{

public static int oddSumcheck(int[] arr, int size)

{

int sum = 0;

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

{

if (i % 2 != 0)

{

sum = sum + arr[i];

}

}

if (sum % 2 == 0)

{

return 1;

}

else

{

return 0;

}

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace OddEvenSum

{

class Program

{

static void Main(string[] args)

{

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

string input = Console.ReadLine();

string[] ip2 = input.Split(' ');

int[] arr = new int[size];

for (int j = 0; j < size; j++)

{

arr[j] = int.Parse(ip2[j]);

}

int res=UserMainCode.oddSumcheck(arr, size);

if (res == 1)

{

Console.WriteLine("Even");

}

else if (res == 0)

{

Console.WriteLine("Odd");

}

}

}

}

Q6) TakeHomeSalary

UserMainCode.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace TakeHomeSalary

{

class UserMainCode

{

public static int calculatesalary(int sal)

{

int pf = 0;

if (sal <= 15000)

{

pf = 750;

}

else if (sal >= 15001 || sal <= 22000)

{

pf = 850;

}

else if (sal >= 22001 || sal <= 30000)

{

pf = 925;

}

else if (sal > 33000)

{

pf = 1000;

}

else

{

return -1;

}

return sal - pf - 678;

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace TakeHomeSalary

{

class Program

{

static void Main(string[] args)

{

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

int res = UserMainCode.calculatesalary(salary);

if (res != -1)

{

Console.WriteLine(res);

}

}

}

}

1>Validate ID Location:  
Input:  
Input1:CTS-hyd-1234  
Input2:hyderabad  
Output:Valid

UserMainCode.cs  
using System.Text;  
using System.Threading.Tasks;  
using System.Text.RegularExpressions;  
namespace Validate\_ID  
{  
    class UserMainCode  
    {  
        public static int validateIDLocation(string s1, string s2)  
        {  
            int output = 0;  
            string pattern = "[CTS]+[-]+[A-Za-z]{3}[-]+[0-9]{4}";  
            bool res = Regex.IsMatch(s1,pattern);  
            if (res)  
            {  
                //string check = s2.ToUpper();  
                if (s1.Contains(s2.Substring(0, 3)))  
                {  
                    output = 1;  
                }

                else  
                {  
                    output = -1;  
                }  
            }  
            else {   
                output = -2;   
            }  
            return output;  
        }  
    }  
}

Program.cs  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;

namespace Validate\_ID  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
              
            string s1=Console.ReadLine();  
            string s2=Console.ReadLine();  
           int i = UserMainCode.validateIDLocation(s1,s2);  
            if (i == 1)  
            {  
                Console.WriteLine("Valid");  
            }  
            else   
            {  
                Console.WriteLine("Invalid");  
            }  
        }  
    }  
}

------------------------------------------------------------------------------------------------------------------------------------------------------------------  
2> Electric Bill

input1 =ABC2012345  
input2 = ABC2012660  
input3 = 4  
Bill = (12660 - 12345 ) \* 4  
output = 1260

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

namespace Calculate\_Bill  
{  
    class UserMainCode  
    {  
        public static int calBill(string s1, string s2, int no)  
        {  
            int sum = 0;  
            string b1 = s1.Substring(5, 5);  
            string b2 = s2.Substring(5, 5);  
            int a1 = int.Parse(b1);  
            int a2 = int.Parse(b2);  
            sum = (a2 - a1) \* 4;  
            return sum;  
        }  
    }  
}

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

namespace Calculate\_Bill  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            string str1 = Console.ReadLine();  
            string str2 = Console.ReadLine();  
            int no = int.Parse(Console.ReadLine());  
            int res = UserMainCode.calBill(str1, str2, no);  
            Console.WriteLine(res);  
        }  
    }  
}

3>EMI Calculation  
Input:  
(date)-->01-11-1983  
(month)-->12  
BL:i)month should be 12 ,24, 36, 48 ,otherwise print -2.  
ii)if date is not in correct format ,print -1.  
Output:26250

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

namespace Calculate\_EMI  
{  
    class UserMainCode  
    {  
        public static int emi(string dob, int month)  
        {  
            if ((month != 12) && (month != 24) && (month != 36) && (month != 48))  
            {  
                return -2;  
            }  
            DateTime dt1;  
            int inst = 0;  
            double amount;  
            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;  
                DateTime dt3 = DateTime.Now.Date;  
                DateTime dt4 = dt3.AddMonths(month);  
                if (mont < 0)  
                {  
                    year = year - 1;  
                    mont = mont + 12;  
                }  
                if (year <= 22)  
                {  
                    amount = (double)200000 \* 1.03;  
                    inst = (int)amount / month;  
                }  
                if (year > 22 && year <= 45)  
                {  
                    amount = (double)300000 \* 1.05;  
                    inst = (int)amount / month;  
                }  
                if (year > 45 && year <= 100)  
                {  
                    amount = (double)500000 \* 1.07;  
                    inst = (int)amount / month;  
                }  
                return inst;  
            }  
            else  
            {  
                return -1;  
            }  
        }  
    }  
}

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

namespace Calculate\_EMI  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            string s = Console.ReadLine();  
            int n = int.Parse(Console.ReadLine());  
            int c = UserMainCode.emi(s, n);  
            Console.WriteLine(c);  
        }  
    }  
}

4>Remove Tens  
Remove tens from the array and shift elements towards left and fill the trailing space by 0.  
Input the array as ,size of array followed by elements of array.  
Input:  
5  
1  
10  
20  
10  
2  
Output:  
1  
20  
2  
0  
0

5>Sum of squares of elements at even position and cubes of elements at odd position  
Input consist of size of array followed by array elements.  
Input:  
3  
1  
2  
3  
Output:  
18

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

namespace sum\_of\_squares\_and\_cubes  
{  
    class UserMainCode  
    {  
        public static int sum(int n, int[] a)  
        {  
            int esum = 0,osum=0,total=0;  
            for (int i = 0; i < n; i++)  
            {  
                if (i % 2 == 0)  
                {  
                    esum = (int)(esum+Math.Pow(a[i],2));  
                }  
                else  
                {  
                    osum = (int)(osum + Math.Pow(a[i], 3));  
                }  
            }  
            total = esum + osum;  
            return total;  
        }  
    }  
}

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

namespace sum\_of\_squares\_and\_cubes  
{  
    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 res = UserMainCode.sum(n,arr);  
            Console.WriteLine(res);  
        }  
    }  
}

6>Calculate Cost  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
class UserMainCode  
{  
    public static int getCost(int cost, char fltype, char dctype)  
    {  
        int final = 0;  
        if (fltype == 'N' || fltype == 'E')  
        {  
            if (fltype == 'N')  
            {  
                final = 400 \* cost;  
            }  
            else if (fltype == 'E')  
            {  
                final = 700 \* cost;  
            }  
            if (dctype == 'S' || dctype == 'C')  
            {  
                if (dctype == 'S')  
                {  
                    final = final + 15000;  
                }  
                else if (dctype == 'C')  
                {  
                    final = final + 25000;  
                }  
            }  
            else  
            {  
                return -3;  
            }  
        }  
        else  
        {  
            return -2;  
        }

        if (final < 20000)  
        {  
            return -1;  
        }  
        else  
        {  
            return final;  
        }  
    }  
}

using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
class Program  
{  
    static void Main(string[] args)  
    {  
        int cost = int.Parse(Console.ReadLine());  
        char fltype = char.Parse(Console.ReadLine());  
        char dctype = char.Parse(Console.ReadLine());  
        UserMainCode umc = new UserMainCode();  
        int result = UserMainCode.getCost(cost, fltype, dctype);  
        if (result == -2)  
        {  
            Console.WriteLine("Invalid flower type");  
        }  
        else if (result == -3)  
        {  
            Console.WriteLine("Invalid decoration type");  
        }  
        else if (result == -1)  
        {  
            Console.WriteLine("Too low cost");  
        }  
        else  
        {  
            Console.WriteLine(result);  
        }  
    }  
}

1.) Duplicate Date Elements

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Globalization;

namespace tripprob1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter number of dates you want to check");

int num = Int32.Parse(Console.ReadLine());

string[] st1 = new string[num];

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

{

st1[i]=Console.ReadLine();

}

Program p = new Program();

p.Display(st1);

Console.ReadKey();

}

public void Display(string[] st)

{

DateTime d,d1, d2, d3;

string[] res = new string[st.Length];

bool val = false;

int k = 0;

foreach(var c in st)

{

//d = DateTime.ParseExact(st[0], "dd-MM-yyyy", null);

if (DateTime.TryParseExact(c, "dd-MM-yyyy", null, DateTimeStyles.None, out d))

{

Console.WriteLine(d);

}

else if (DateTime.TryParseExact(c, "dd/MM/yyyy", null, DateTimeStyles.None, out d1))

{

Console.WriteLine(d1);

}

else if (DateTime.TryParseExact(c, "dd.MM.yyyy", null, DateTimeStyles.None, out d2))

{

Console.WriteLine(d2);

}

else if (DateTime.TryParseExact(c, "dd/MMMM/yyyy", null, DateTimeStyles.None, out d3))

{

Console.WriteLine(d3);

}

else

{

Console.WriteLine("Invalid");

val = true;

}

if (!val)

{

res[k] = c;

k++;

val = false;

}

}

foreach (var x in res.Distinct())

{

Console.WriteLine(x);

}

}

}

}

2.) Remove Duplicates

program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace tripprob2

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter string");

string st = Console.ReadLine();

string fs= UserProgrameCode.removeDuplicates(st);

Console.WriteLine(fs);

Console.ReadKey();

}

}

}

UserProgramCode.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace tripprob2

{

class UserProgrameCode

{

public static string removeDuplicates(string st)

{

char[] res = new char[st.Length \* 2];

int i = 0;

foreach (var c in st)

{

if (c == ' ' || (!res.Contains(c)))

{

res[i] = c;

i++;

}

}

string ret = new string(res);

return ret;

}

}

}

3.) count Vowels

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace tripprob4

{

class Program

{

static void Main(string[] args)

{

do

{

UserProgramCode p = new UserProgramCode();

Console.WriteLine(" Enter string to count Vowels");

string st = Console.ReadLine();

int ans = p.countVowels(st);

Console.WriteLine(ans);

Console.ReadKey();

} while (true);

}

}

}

UserProgramCode.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace tripprob4

{

class UserProgramCode

{

public int countVowels(string st)

{

st.ToLower();

int count = 0;

if (!st.All(char.IsLetter))

{

//Console.WriteLine("Input contains digit/special Symbols also");

count = -1;

goto finish;

}

char[] arr = st.ToCharArray();

foreach (var c in arr)

{

if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u')

{

count++;

}

}

finish:

return count;

}

}

}

5.) Relative Order

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace relative\_order

{

class Program

{

static void Main(string[] args)

{

int n1 = Convert.ToInt32(Console.ReadLine());

int check1 = 0;

int[] a1 = new int[n1];

for (int i = 0; i < n1; i++)

{

a1[i] = Convert.ToInt32(Console.ReadLine());

if (a1[i] < 0)

check1++;

}

if (a1.Length < 3 || a1.Length > 15)

{

Console.WriteLine("-3");

}

int n2 = Convert.ToInt32(Console.ReadLine());

int check2 = 0;

int[] a2 = new int[n2];

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

{

a2[i] = Convert.ToInt32(Console.ReadLine());

if (a2[i] < 0)

check2++;

}

if (check1 > 0 || check2 > 0)

{

Console.WriteLine("-1");

Console.ReadKey();

Environment.Exit(0);

}

string s1 = null;

foreach (int x in a1)

{

s1 = string.Concat(s1, x.ToString());

}

string s2 = null;

foreach (int x in a2)

{

s2 = string.Concat(s2, x.ToString());

}

string ans = null;

//Console.WriteLine(s1);

for (int i = 0; i < s2.Length; i++)

{

char c2 = s2[i];

int index = s1.IndexOf(c2);

if (index >= 0)

{

while (index >= 0)

{

ans = string.Concat(ans, s1[index]);

s1 = s1.Remove(index, 1);

index = s1.IndexOf(c2);

}

}

else

{

Console.WriteLine("-2");

}

}

Array.Sort(a1);

int z = 0;

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

{

string c1 = a1[i].ToString();

int index2 = ans.IndexOf(c1);

if (index2 == -1)

{

ans = string.Concat(ans, a1[i].ToString());

}

}

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

{

Console.WriteLine(ans[i]);

}

Console.ReadKey();

}

}

}

OR

UserProgramCode.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace tripprob5

{

class UserProgramCode

{

public int[] mechanism(int[] large, int[] small)

{

int[] res= new int[large.Length];

int[] res2 = new int[large.Length];

int[] fi = new int[large.Length];

int i = 0;

int j = 0;

foreach (var c in small)

{

foreach (var x in large)

{

if (c == x)

{

res[i] = x;

i++;

}

}

}

foreach (var x in res)

{

foreach (var z in large)

{

if (x != z && (!res2.Contains(z)) && (!res.Contains(z)))

{

res2[j] = z;

j++;

}

}

}

int s = 0;

foreach (var z in res)

{

if (z != 0)

{

fi[s] = z;

s++;

}

}

foreach (var m in res2)

{

if (m != 0)

{

fi[s] = m;

s++;

}

}

return fi;

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace tripprob5

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter size of first string");

int size=Int32.Parse(Console.ReadLine());

int[] s= new int[size];

Console.WriteLine("Enter elements");

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

s[i]=Int32.Parse(Console.ReadLine());

Console.WriteLine("Enter size of first string");

int size2=Int32.Parse(Console.ReadLine());

int[] s2= new int[size2];

Console.WriteLine("Enter elements");

for(int i=0;i<size2;i++)

s2[i]=Int32.Parse(Console.ReadLine());

UserProgramCode p = new UserProgramCode();

int[] res= p.mechanism(s,s2);

foreach(var c in res)

{

Console.WriteLine(c);

}

Console.ReadKey();

}

}

}

6.) Decimal To Roman

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace tripprob6

{

class Program

{

static void Main(string[] args)

{

do

{

Console.WriteLine("Enter number");

int st = Int32.Parse(Console.ReadLine());

StringBuilder sb = new StringBuilder();

int[] num = { 1, 4, 5, 9, 10, 40, 50, 90, 100, 400, 500, 900, 1000 };

string s1 = "";

string[] s = { "I", "IV", "V", "IX","X", "XL", "L", "XC", "C", "CD", "D", "CM", "M" };

while (st > 0)

{

for (int i = num.Length - 1; i >= 0; i--)

{

if (st / num[i] >= 1)

{

st = st - num[i];

sb.Append(s[i]);

break;

}

}

}

Console.WriteLine(sb);

Console.ReadKey();

} while (true);

}

}

}

87)87.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  
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;  
}  
}

6262.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  
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;  
}  
}

validate id location  
34.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

13.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

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";  
Console.ReadKey();  
}  
}  
class Program  
{  
static void Main(string[] args)  
{  
string str = Console.ReadLine();  
Console.WriteLine(UserProgramCode.validateNumber(str));  
}  
}  
}

Sum of digit in string:

Eg:

Enter the string :jui123hf3

Sum of digit :9

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace sum\_of\_digit\_in\_string

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter the string (eg:jui23g4):");

string s = Console.ReadLine();

char[] c = s.ToCharArray();

int i;

int sum = 0;

for (i = 0; i < s.Length; i++)

{

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

{

int x= Convert.ToInt32(c[i].ToString());

sum = sum + x;

}

}

Console.WriteLine("Sum of digits in string is:");

Console.WriteLine(sum);

Console.ReadKey();

}

}

}

Eg:

Enter string: Cognizant Technology Solutions LTD

String 1: Technology

String 2: LTD

Yes

Enter string: Cognizant Technology Solutions LTD

String 1: LTD

String 2: Technology

No

Enter string: Cognizant Technology Solutions LTD

String 1: LTD

String 2: pvt

No

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace myContains

{

class Program

{

static void Main(string[] args)

{

string s = Console.ReadLine();

string s1 = Console.ReadLine();

string s2 = Console.ReadLine();

if (s.Contains(s1) && s.Contains(s2))

{

if (s.IndexOf(s1) < s.IndexOf(s2))

{

Console.WriteLine("yes");

}

else

{

Console.WriteLine("no");

}

}

else { Console.WriteLine("no"); }

Console.ReadKey();

}

}

}

Largest span of 1:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LargestSpan

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine(“enter size of array:”);

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

int[] a = new int[n];

int i;

Console.WriteLine(“enter elements of array:”);

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

{

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

}

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

int x = 0, b = 0, c = 0;

foreach (int item in a)

{

x = Array.IndexOf(a, item);

b = Array.LastIndexOf(a, item);

c = (b - x) + 1;

li.Add(c);

}

if (li.Count == 0)

Console.WriteLine(0);

else

Console.WriteLine(“span:”+li[0]);

Console.ReadKey();

}

}

}

1)Add the non-equal elements in the two arrays

Input:

7

1

2

3

4

5

6

7

3

4

5

6

Output:

13(1+2+3+7)

Code:

UserMainCode.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Addnon

{

class UserMainCode

{

public static int checkSum(int size1, int[] a, int size2, int[] b)

{

int temp = 0, temp1 = 0, sum = 0;

for (int i = 0; i < size1; i++)

{

if (a[i] < 0)

{

temp = -1;

}

}

for (int i = 0; i < size2; i++)

{

if (b[i] < 0)

{

temp1 = -2;

}

}

if (temp == -1 && temp1 == -2)

{

return -3;

}

else if (temp == -1 && temp1 == 0)

{

return -1;

}

else if (temp == 0 && temp1 == -2)

{

return -2;

}

else

{

int[] total = new int[size1 + size2];

int k = size1;

for (int i = 0; i < size1; i++)

{

total[i] = a[i];

}

for (int i = 0; i < size2; i++)

{

total[k++] = b[i];

}

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

{

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

{

if (total[i] == total[j])

{

total[j] = 0;

total[i] = 0;

}

}

}

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

{

sum += total[i];

}

return sum;

}

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Addnon

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter the size of the first array:");

int size1=Convert.ToInt32(Console.ReadLine());

int[] a = new int[size1];

Console.WriteLine("Enter the size of the second array:");

int size2 = Convert.ToInt32(Console.ReadLine());

int[] b = new int[size2];

Console.WriteLine("Enter the elements of the first array:");

for (int i = 0; i < size1; i++)

{

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

}

Console.WriteLine("Enter the elements of the second array:");

for (int i = 0; i < size2; i++)

{

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

}

int x=UserMainCode.checkSum(size1,a,size2,b);

if(x==-3)

Console.WriteLine("Both the arrays have invalid input");

else if (x==-1)

{

Console.WriteLine("Array1 has invalid input");

}

else if (x==-2)

{

Console.WriteLine("Array2 has invalid input");

}

else

Console.WriteLine("The sum of the non equal elements are:{0}", x);

}

}

}

Q2)

Write a program to encrypt a given password with the letter given.If the first letter of the password

matches the letter given then replace it with the next letter followed by #.

Input

Red Apple

G

Ouput:

Ged Gpple

Input:

Red Apple

R

Ouput:

S#ed Rpple

Code:

UserPrograCode.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Password

{

class UserProgramCode

{

public static string replaceChar(string a, string b)

{

StringBuilder sb1 = new StringBuilder();

string h = " ";

string u = "!@#$%^&\*()";

//if (b == h||b==u)

//{

// sb1.Append("Invalid password:");

// return sb1.ToString();

//}

char[] v = a.ToCharArray();

//if (char.IsDigit(nv))

//{

// sb1.Append("Invalid");

// return sb1.ToString();

//}

foreach (var p in v)

{

if (char.IsDigit(p))

{

sb1.Append("Invalid password:");

return sb1.ToString();

}

for (int z = 0; z < u.Length; z++)

{

if (p == u[z])

{

sb1.Append("Invalid password:");

return sb1.ToString();

}

}

}

StringBuilder sb = new StringBuilder();

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

foreach (string s in r)

{

int t = s.Length;

char[] ch = s.ToCharArray();

if (s.Substring(0, 1) == b)

{

char q = (char)(ch[0] + 1);

string n = s.Substring(1, t - 1);

sb.Append(q);

sb.Append('#');

sb.Append(n);

sb.Append(' ');

}

else if (s.Substring(0, 1) != b)

{

string p = s.Substring(1, t - 1);

sb.Append(b);

sb.Append(p);

sb.Append(' ');

}

}

return sb.ToString();

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Password

{

class Program

{

static void Main(string[] args)

{

string in1 = Console.ReadLine();

string in2 = Console.ReadLine();

string op = UserProgramCode.replaceChar(in1, in2);

Console.WriteLine(op);

}

}

}

Q3)Write a program to rea a String and a character and to reverse the string and return

it in a format such that each character is seperated by the given character.Print the final

string.

Input:

Rabbit

Output:

t-i-b-b-a-R

Code:

UserProgramCode.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Revstr

{

class UserProgramCode

{

public static string rev(string str, char c)

{

StringBuilder sb = new StringBuilder();

char[] ch = str.ToCharArray();

Array.Reverse(ch);

for (int i = 0; i < ch.Length - 1; i++)

{

sb.Append(ch[i]);

sb.Append(c);

}

sb.Append(ch[ch.Length - 1]);

string s = sb.ToString();

return s;

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Revstr

{

class Program

{

static void Main(string[] args)

{

string str = Console.ReadLine();

char c = Convert.ToChar(Console.ReadLine());

Console.WriteLine(UserProgramCode.rev(str, c));

}

}

}

Q4)Write a program to check if the element in a certain position is same in the given two

strings or not. If the position is n then traverse string 1 from first to the nth position and

in string 2 traverse from the back to the nth position and check if the element is same.

Input:

Battle

Final

2

Output:

The character in a is same.

Input:

Red

Hello

3

Ouput:

The character d and l are not same.

Code:

UserProgramCode:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Checkcharacter

{

class UserProgramCode

{

public static string check(string a, string b, int n)

{

char[] ch1 = a.ToCharArray();

char[] ch2 = b.ToCharArray();

Array.Reverse(ch2);

StringBuilder sb = new StringBuilder();

if (ch1[n - 1] == ch2[n - 1])

{

sb.Append("The character ");

sb.Append(ch1[n - 1]);

sb.Append(" is same.");

}

else

{

sb.Append("The character ");

sb.Append(ch1[n - 1]);

sb.Append(" is not same as ");

sb.Append(ch2[n - 1]);

sb.Append(".");

}

return sb.ToString();

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Checkcharacter

{

class Program

{

static void Main(string[] args)

{

string str1 = string.Empty;

string str2 = string.Empty;

int n = 0;

Console.WriteLine("Enter the first string:");

str1 = Console.ReadLine();

Console.WriteLine("Enter the second string:");

str2 = Console.ReadLine();

Console.WriteLine("Enter the position:");

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

string s = UserProgramCode.check(str1, str2, n);

Console.WriteLine(s);

}

}

}

Q5)Write a program to sort a string array in descending order

Input:

Apple

Mango

Blueberry

Output:

Blueberry

Apple

Mango

Code:

UserProgramCode.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Strsort

{

class UserProgramCode

{

public static void Checksort(string[] a)

{

var x = from z in a orderby z.Length descending select z;

Console.WriteLine("The sorted array is:");

foreach (var v in x)

{

Console.WriteLine(v);

}

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Strsort

{

class Program

{

static void Main(string[] args)

{

int size1 = 0;

Console.WriteLine("Enter the size:");

size1 = Convert.ToInt32(Console.ReadLine());

string[] str = new string[size1];

Console.WriteLine("Enter the elements:");

for (int i = 0; i < size1; i++)

{

str[i] = Console.ReadLine();

}

UserProgramCode.Checksort(str);

}

}

}

Q6)A student 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 Rules:

(a) He/She should get 55% or more in A and 45% or more in B

(b) if He/She gets less than 55% in A he/she should get 55% in B.

However he/she should get atleast 45% in A.

(c)If he/she gets less than 45% in B he/she and 65% or more 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 the above rules.

Code:

UserProgramCode.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Marks

{

class UserProgramCode

{

public static string findStatus(int a, int b)

{

if (a > 100 || b > 100)

return "Invalid Input";

else if (a >= 55 && b >= 45)

return "P";

else if (a >= 45 && b >= 55)

return "P";

else if (b < 45 && a >= 65)

return "R";

else

return "F";

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Marks

{

class Program

{

static void Main(string[] args)

{

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

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

string r = UserProgramCode.findStatus(m, n);

Console.WriteLine(r);

}

}

}

1.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;  
}

2.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 ('Ro' 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 rate  
{  
       class Program  
{  
            static void Main(string[] args)  
{  
            int n = int.Parse(Console.ReadLine());  
            char ch = char.Parse(Console.ReadLine());  
              Console.WriteLine(UserCode.CalcShippingCost(n,ch));  
}  
}  
}  
           class UserCode  
{  
              public static double CalcShippingCost(int gms, char ch)  
{  
double charge;  
if (ch == 'R' && gms <= 2000)  
{  
charge = (gms \* .25);  
}  
else if (ch == 'R' && gms >= 2000)  
{  
charge = (gms - 2000) \* .35 + (2000 \* .25);  
}  
else if (ch == 'E' && gms <= 2000)  
{  
charge = (gms \* .25) + 50;  
}  
else if (ch == 'E' && gms >= 2000)  
{  
charge = (gms - 2000) \* .35 + (2000 \* .25) + 50;  
}  
else  
{  
return 0;  
}  
return charge;  
}  
}

3.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  
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;  
}  
}

.  
4.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;  
}  
}  
}  
}  
===================================================================

 5.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  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
namespace program33  
{  
class Program  
{  
static void Main(string[] args)  
{  
int n;  
n= UserProgramCode.countcharacters("qwerty");  
Console.WriteLine(n);  
}  
}  
class UserProgramCode  
{  
public static int countcharacters(string str)  
{  
int len = str.Length;  
return len;  
}  
}  
}

6.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

namespace flower   
{

class UserMainCode  
{   
public static int fun(int cost, char dtype, char ftype)   
{

     int total;  
    if (dtype == 'S')   
{   
         if (ftype == 'N')    
{          total = 15000 + (cost \* 400);   
      } else    
        if (ftype == 'E')    
{          total = 15000 + (cost \* 700);   
    }       
 else     
         {   
        total = -2; return total;   
          }   
           }   
         else if (dtype == 'C')  
             {  
              if (ftype == 'N')   
                {   
               total = 25000 + (cost \* 400);  
                    }   
                else if (ftype == 'E')  
                 {   
                  total = 25000 + (cost \* 700);   
                    
                 } else   
                  
             {   
                    total = -2;   
                   
                 return total;   
                }   
              }    
                else   
                   
               {   
                 total = -3;  
                    
               return total;   
                              }  
                      if (total < 20000)

                           {   
                            total = -1;   
                       }   
   
                          return total;   
}   
  } }  
                    public static void count(int n,char a,char b)   
           
              {   
                      int flower;   
                             
                           if (a == 'n')  
                        {   
                                 flower = n \* 400;   
                          }   
                              else   
                                   {  
                                         flower = n \* 700;   
                                     } if (b == 's')  
                                              
                                       {

                                       flower = flower + 15000;  
                               
                          }

                                    else

                               {

                             flower = flower + 25000;

                       }

                        if (flower < 20000)   
                         
                    {   
                           Console.WriteLine(-1);  
                     }   
                    else   
                     {   
                                Console.WriteLine(flower);   
                      }  
                          }

7.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

8.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

public static void user(string a)

{

      Regex reg = new Regex(@"^([A-Z]{3})+([0-9]{2})+[DN]+([0-9]{3})$");

         if (reg.IsMatch(a))

        {   
           string st1=a.Substring(0,3);   
           
             string st2="";   
         
               string st3 = a.Substring(3, 2);  
                
                string st4;

                   string st5 = a.Substring(5, 2);   
   
                      string st6;   
         
                   string st7 = a.Substring(7, 3);

                   string st8;

                  if (st1 == "HYD") st2 = "hyderabad";

                      else if (st1 == "CBE") st2 = "coimbatore";   
       
                         else if (st1 == "KOL") st2 = "kolkatta";   
   
                         else if (st1 == "PUN") st2 = "pune";   
                 
                         else if (st1 == "BGL") st2 = "banglore";   
   
                           else if (st1 == "CHN") st2 = "chennai";

                           else if (st1 == "KOC") st2 = "kochi";

                            else { Console.WriteLine("no");  
      
                                Environment.Exit(0);   
         
                         }

                            Console.WriteLine(" Dotnet batch {0} has joined in 20{1} year and is at {2} location",st7,st3,st2);

}  
                          else Console.WriteLine("no");  
 }  
------------------------------------------------------------------------------------------------------------------------------

                       class UserProgramCode   
                               {

                              public static string checkBatch(string input1)

                               {   
                       // fill code here string ans;

                         char[] ch = input1.ToCharArray();

                        string loc = input1.Substring(0, 3);  
       
                      if (loc != "CHN" && loc != "CBE " && loc != "KOC" && loc != "PUN" && loc != "BGL" && loc != "HYD" && loc != "KOL")

                        {  
                           ans = "-1";  
                         return ans;

                                  }  
                          string yr = input1.Substring(3, 2);   
                              string bat = input1.Substring(7);   
                              string pr=null;

                                if(!(char.IsDigit(ch[3]) && char.IsDigit(ch[4]) && char.IsDigit(ch[7]) && char.IsDigit(ch[8])   
                                      && char.IsDigit(ch[9])))

                                       {   
                                         ans = "-2";   
                               return ans;   
                                        }

                                                    string dn=input1.Substring(5,2);   
                                      
                                                  if(!(dn=="DN"))   
                                                    {

                                                   ans = "-3";  
                           
                                                       return ans;  
                                                    }   
                                            if(loc=="CHN") pr="Chennai";   
                                      else if(loc=="CBE") pr="Coimbatore";   
   
                                     else if(loc=="KOC") pr="Kochi";   
                                      else if(loc=="PUN") pr="Pune";   
                               else if(loc=="BGL") pr="Bangalore";  
                                 else if(loc=="HYD") pr="Hyderabad";   
                              else if(loc=="KOL") pr="Kolkata";   
                                ans="DotNet batch "+bat+" has joined in 20"+yr+" year and is at "+pr+" Location";  
                                return ans;

                          }    
                             }  
                                Close

**grade**

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 namespace level2csharp {

class Program

{ static void Main(string[] args)

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

int[] a = new int[n]; string gr = ""; for (int i = 0; i < n; i++)

{ a[i] = int.Parse(Console.ReadLine()); } int max = 0,rollno=0;

for (int i = 0; i < n; i=i+2)

{ if (a[i] > max)

{ max = a[i]; rollno = a[i + 1]; } } if (max >= 80) { gr = "DISTINCTION"; }

else if (max >= 60 && max < 80) { gr = "FIRST CLASS"; }

else if (max >= 45 && max < 60)

{ gr = "SECOND CLASS"; }

Console.WriteLine(rollno+" "+"has"+" "+"scored"+gr); } } }

Close

**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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace insurance

{

class Program

{

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;

namespace insurance

{

class Userprogramcode

{

public static void InsuranceGuide(char health, int age, char gender, char location)

{

if (health == 'E')

{

if (age >= 25 && age <= 35)

{

if (location == 'C')

{

if (gender == 'M')

{

Console.WriteLine("4");

Console.WriteLine("200000");

}

else

{

Console.WriteLine("3");

Console.WriteLine("100000");

}

}

else

{

Console.WriteLine("The person cannot be insured");

}

}

else

{

Console.WriteLine("Age limit Exceeded");

}

}

else

{

if (age >= 25 && age <= 35)

{

if(location == 'V')

{

if (gender == 'M')

{

Console.WriteLine("6");

Console.WriteLine("10000 ");

}

}

}

else

{

Console.WriteLine("The person cannot be insured");

}

}

}

}

}

**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

namespace flower

{

class UserMainCode

{

public static int fun(int cost, char dtype, char ftype)

{

int total;

if (dtype == 'S')

{

if (ftype == 'N')

{

total = 15000 + (cost \* 400);

}

else if (ftype == 'E')

{

total = 15000 + (cost \* 700);

}

else

{

total = -2;

return total;

}

}

else if (dtype == 'C')

{

if (ftype == 'N')

{

total = 25000 + (cost \* 400);

}

else if (ftype == 'E')

{

total = 25000 + (cost \* 700);

}

else

{

total = -2;

return total;

}

}

else

{

total = -3;

return total;

}

if (total < 20000)

{

total = -1;

}

return total;

}

}

}

public static void count(int n,char a,char b)

{

int flower;

if (a == 'n')

{

flower = n \* 400;

}

else

{

flower = n \* 700;

}

if (b == 's')

{

flower = flower + 15000;

}

else

{

flower = flower + 25000;

}

if (flower < 20000)

{

Console.WriteLine(-1);

}

else

{

Console.WriteLine(flower);

}

}

**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

namespace vowel

{

class Program

{

static void Main(string[] args)

{

string s= Console.ReadLine();

StringBuilder sb = new StringBuilder();

int count = 0;

char[] c = s.ToCharArray();

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

{

if (c[i] == 'a' || c[i] == 'e' || c[i] == 'i' || c[i] == 'o' || c[i] == 'u')

{

sb.Append(c[i]);

count++;

}

}

if(count==5)

Console.WriteLine("yes");

else

Console.WriteLine("no");

}

}

**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

namespace oddeven

{

class Program

{

static void Main(string[] args)

{

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

int[] arr=new int[50];

int esum = 0, osum = 0,rem=0,k=0;

while (n > 0)

{

rem = n % 10;

arr[k] = rem;

k++;

n = n / 10;

}

Array.Resize(ref arr,k);

Array.Reverse(arr);

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

{

if (i % 2 == 0)

esum = esum + arr[i];

else

osum = osum + arr[i];

}

if (osum == esum)

{

Console.WriteLine(1);

}

else

{

Console.WriteLine(-1);

Console.WriteLine(osum);

Console.WriteLine(esum);

}

}

}

}

Find Leader

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace E1

{

class Program

{

static void Main(string[] args)

{

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

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

int[] a = new int[n];

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

{

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

}

List<int> o = UserProgram.findLeader(a);

foreach (var x in o)

{

Console.WriteLine(x);

}

Console.ReadKey();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace E1

{

class UserProgram

{

public static List<int> findLeader(int []a)

{

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

for (int i = 0; i < a.Length-1; i++)

{

int f=0;

for (int j = i + 1; j < a.Length; j++)

{

if (a[i] == a[j])

{

li.Add(-3);

return li;

}

if (a[i] < 0)

{

li.Add(-1);

return li;

}

if (a[i] < 2 || a[i] > 10)

{

li.Add(a[i]);

return li;

}

if (a[i] < a[j])

{

f = 1;

break;

}

}

if (f == 0)

{

li.Add(a[i]);

}

}

li.Add(a[a.Length - 1]);

li.Sort();

return li;

}

}

}

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

Find Charge

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace e3

{

class Program

{

static void Main(string[] args)

{

string a = Console.ReadLine();

string b = Console.ReadLine();

int c=UserProgram.findCharge(a, b);

Console.WriteLine(c);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace e3

{

class UserProgram

{

public static int findCharge(string a, string b)

{

DateTime d3;

DateTime d4;

bool res = DateTime.TryParseExact(a, "yyyy-MM-dd:HH:mm:ss",null, System.Globalization.DateTimeStyles.None, out d3);

bool res1 = DateTime.TryParseExact(b, "yyyy-MM-dd:HH:mm:ss", null, System.Globalization.DateTimeStyles.None, out d4);

if (res == false || res1 == false)

{

return -1;

}

if (d3 > d4)

{

return -2;

}

double c = d3.Subtract(d4).TotalHours;

if (c > 24)

{

return -3;

}

int amt = 0;

if (c <= 3)

{

amt=20;

}

else if (c <= 24 && c > 3)

{

amt = 20 + ((int)c - 3) \* 5;

}

return amt;

}

}

}

1>

Enter the int array and add the smallest and largest number and print the reverse of the sum.

Input : {3 9 5 10}

Output : 31

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace addreverse

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter the n:");

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 sum = UserProgrameCode.Addandreverse(arr);

Console.Write("The sum is:"+sum);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace addreverse

{

class UserProgrameCode

{

public static int Addandreverse(int [] a)

{

int sum=0;

Array.Sort(a);

int sum1 = a[0]+a[a.Length-1];

while (sum1 > 0)

{

int x = sum1 % 10;

sum = sum \* 10 + x;

sum1 = sum1 / 10;

}

return sum;

}

}

}

2) Validate voter:

using System.Text;

using System.Threading.Tasks;

namespace Age

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter the Date of birth(dd/MM/yyyy):");

string dob = Console.ReadLine();

Console.WriteLine("Enter the Date of election(dd/MM/yyyy):");

string edate = Console.ReadLine();

int res = UserProgramCode.validateVoter(dob,edate);

if (res == -1)

Console.WriteLine("not Eligible!!");

else if (res == 0)

Console.WriteLine("Eligible!!");

else

Console.WriteLine("Invalid");

}

}

}

using System;

using System.Text.RegularExpressions;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Globalization;

namespace Age

{

class UserProgramCode

{

public static int validateVoter(string dob, string edate)

{

DateTime dt1,dt2;

bool d1 = DateTime.TryParseExact(dob, "dd/MM/yyyy",CultureInfo.InvariantCulture,DateTimeStyles.None, out dt1);

bool d2 = DateTime.TryParseExact(edate,"dd/MM/yyyy", CultureInfo.InvariantCulture,DateTimeStyles.None, out dt2);

if(d1==true && d2==true)

{

TimeSpan ts = dt2.Subtract(dt1);

int age = ts.Days / 365;

//Console.WriteLine(age);

if (age >= 18)

return (0);

else

return (-1);

}

else

return (1);

}

}

}

6363.Sum of cubes and squares of elements in an 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

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;

}

}

===========

15.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;

using System.Text.RegularExpressions;

namespace mock\_pgm2

{

public class Isletter

{

public bool IsAlphaNumeric(string input)

{

return Regex.IsMatch(input, "^[a-zA-Z]+$");

}

}

class Program

{

static void Main(string[] args)

{

Isletter obj=new Isletter();

int count = 0, temp = 0 , alpha=1;

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

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

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

{

ls.Add(Console.ReadLine());

if (!obj.IsAlphaNumeric(ls[i]))

{

alpha = 0;

}

}

char ch = char.Parse(Console.ReadLine());

if (alpha == 0)

{

Console.WriteLine("String contains non alphabetic characters.");

}

else

{

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

{

if (ls[i].ToCharArray()[0] == char.ToUpper(ch) || ls[i].ToCharArray()[0] == char.ToLower(ch))

{

if (ls[i].Length > count)

count = ls[i].Length;

temp = 1;

}

}

if (temp == 0)

Console.WriteLine("No elements found");

else

{

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

{

if (ls[i].Length == count)

Console.WriteLine(ls[i]);

}

}

}

}

}

}

another method:

using System;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Text;

namespace ConsoleApplication10

{

class Program

{

static void Main(string[] args)

{

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

List<string> list = new List<string>(n);

Int i;

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

list.Add(Console.ReadLine());

char c = char.Parse(Console.ReadLine());

Console.WriteLine(GetlongestString(list, c));

}

static string GetlongestString(List<string> list, char c)

{

string ch = c.ToString();

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

{

char[] c1 = list[i].ToCharArray();

foreach (char c2 in c1)

{

if (!char.IsLetter(c2))

{

return "Non Alphabets Exists";

}

else

{

continue;

}

}

}

var q = from s in list

where s.StartsWith(ch)

orderby s.Length //descending-longest

select s;

foreach (var item in q)

{

return item;

}

return "No Elements Found";

}

}

}

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 Calculate Charge public

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Calulate\_charge

{

class Program

{

static void Main(string[] args)

{

DateTime dt;

DateTime dt1;

double charge=0;

string s = Console.ReadLine();

string s1 = Console.ReadLine();

bool res = DateTime.TryParseExact(s,"yyyy-MM-dd:HH:mm:ss",null,System.Globalization.DateTimeStyles.None,out dt);

bool res1 = DateTime.TryParseExact(s1, "yyyy-MM-dd:HH:mm:ss", null, System.Globalization.DateTimeStyles.None, out dt1);

double hr = dt1.Subtract(dt).TotalHours;

if (res==true && res1==true)

{

if (hr > 24)

Console.WriteLine(-3);

// else if (dt1 > dt)

// Console.WriteLine(-2);

else if (hr <= 3)

charge = 20;

else if (hr > 3 && hr <= 24)

charge = (20 + (hr - 3) \* 5);

Console.WriteLine(charge);

}

else

Console.WriteLine(-1);

Console.ReadKey();

}

}

}

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 ================================================================== ====-======================================================================= Validate String For a given String apply the following validations. 1. The

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Leaders

{

class Program

{

static void Main(string[] args)

{

int c = 0;

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

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

int[] input1=new int[n];

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

input1[i] = Convert.ToInt32(Console.ReadLine());

if (n < 2 || n > 10)

{

Console.WriteLine(-2);

}

else

{

foreach (int p in input1)

{

if (p < 0)

Console.WriteLine(-1);

}

for (int i = 0; i<n-1; i++)

{

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

{

if (input1[i] == input1[j])

Console.WriteLine(-3);

else if (input1[i] > input1[j])

{

}

else

c = 1;

}

if (c == 0)

{

li.Add(input1[i]);

}

c = 0;

}

}

li.Add(input1[input1.Length-1]);

li.Sort();

foreach (var x in li)

{

Console.WriteLine(x.ToString());

}

Console.ReadKey();

}

}

}

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 Password

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace calculate\_Bonus

{

class Program

{

static void Main(string[] args)

{

int bonus;

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

if (sal < 0)

Console.WriteLine(-1);

else if (sal > 1000000)

Console.WriteLine(-2);

else

{

if (sal > 15000 && sal < 20001)

bonus =Convert.ToInt32 ((.17 \* sal) + 1500);

else if(sal>10000 && sal<15001)

bonus = Convert.ToInt32((.15 \* sal) + 1200);

else if(sal<10001)

bonus = Convert.ToInt32((.12 \* sal) + 1000);

else

bonus = Convert.ToInt32((.08 \* sal) + 500);

Console.WriteLine(bonus);

}

Console.ReadKey();

}

}

}

6363.Sum of cubes and squares of elements in an 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

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;

}

}

===================================================================================

**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;

using System.Text.RegularExpressions;

namespace l2ndl3

{

class Program

{

static void Main(string[] args)

{

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

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

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

{

li.Add(Console.ReadLine());

}

char c = char.Parse(Console.ReadLine());

List<string> li2 = new List<string>();

li2 = Usercode.counter(li, c);

foreach (var item in li2)

{

Console.WriteLine(item);

}

}

}

}

{

public static List<string> counter(List<string> li, char c)

{

List<string> q = (from z in li

where !z.StartsWith(c.ToString())

orderby z.Length

select z).ToList();

return q;

}

}