**121.calculate Discount**

Calculate Discount Given two input integer arrays input1 and input2 which contains the details of users booking cars online in the format of key value pairs. input1 (Key,Value) = (UserId1,Bookingmonth1,UserId2,Bookingmonth2,...so on) input2 (Key,Value) = (UserId1,BookingAmount1,UserId2,BookingAmount2,...so on) Write a Program to calculate the discount amount based on the Booking amount and Booking month for every UserId by considering the respective discount rate offered to them and store the result in an output integer array .    
 The discount amount should be calculated using the following specifications ------------------------------------------------------------------------------------------------------------------ Discount Rate                                                           Booking Amount     ------------------------------------------------------------------------------------------------------------------                                                                    |    >=500000        |  >=100000 and < 500000 ------------------------------------------------------------------------------------------------------------------ Discount Rate for Jan-April                     |             25     |            15 Discount Rate for May-August               |            20      |            10 Discount Rate for Sep-December       |  15              |             5 ------------------------------------------------------------------------------------------------------------------ Output should be in the below format: output1 (Key,Value) = (UserId1,DiscountAmount1,UserId2,DiscountAmount2,...so on) Note: 1)Formula: Discount Amount = Booking Amount \* (Discount Rate/100) BUSINESS RULE: 1.If Booking Amount is less than 100000, then discount amount should be zero. 2.If any of the booking month is invalid, then print -1. 3.If any of the booking amount is negative, then print -2. 4.If any of the Booking Month or Booking Amount is missing for any UserId, then print -3 . Create a class named UserProgramCode that has the following static method   
 public static int[] calcDiscount(int[] input1, int[] input2)  
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 The first line of the input consists of an integer, m that corresponds to the number of elements in the input array 1 .  
 The next 'm' lines of input correspond to elements in the input array 1.  
 The next line of the input consists of an integer, n that corresponds to the number of elements in the input array 2 .  
 The next 'n' lines of input correspond to elements in the input array 2.  
 Refer business rules and sample output for formatting specifications.

Sample Input 1 :  
 8

1010 11 1011 02 1012 07 1013 09 8 1010 700000 1011 300000 1012 150000 1013 100000    
 Sample Output 1 : 1010 105000 1011 45000 1012 15000 1013 5000    
 Sample Input 2 : 8 1010 11 1011 02 1012 07 1013 09 6 1010 700000 1011 300000 1012 100000    
 Sample Output 2: -3  

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication7

{

class Program

{

static void Main(string[] args)

{

int m, n,i=0;

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

int[] a = new int[m];

int[] output = new int[m];

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

{

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

}

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

int[] b = new int[n];

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

{

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

}

output = UserProgramCode.calcDiscount(a, b);

if (output[0] < 0)

{

Console.WriteLine(output[0]);

}

else

{

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

{

Console.WriteLine(output[i]);

}

}

Console.Read();

}

}

}

**122.string processing 1**

String Processing I  
 Given a string input input1, write a program to fetch the last n characters from input1 and repeat them after input1 the same number of times as given in the second integer input input2. Business Rules : 1. If the input1 contains any number, print -1. 2. If the input1 contains any special characters, print -2. 3. If the input1 string contains less than input2 number of characters,then print -3.  
 Create a class named UserProgramCode that has the following static method  
 public static string getString(string input1,int input2)  
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 The first line of the input consists of a string and the second line of the input consists of an integer.  
 Refer sample output for formatting specifications.  
 Sample Input 1: Cognizant 3  
 Sample Output 1:  
 Cognizantantantant Sample Input 2:   
 Teach123er 4 Sample Output 2:  
 -1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication13

{

class userprogramcode

{

public static string getString(string ip1, int ip2)

{

string s,final=ip1;

char s1;

for (int i = 0; i < ip1.Length; i++)

{

s1 = ip1[i];

if (char.IsNumber(s1))

return "-1";

else if (!char.IsLetter(s1))

return "-2";

}

s = ip1.Substring(ip1.Length - (ip2));

int j = ip2;

while (j > 0)

{

final = final + s;

j--;

}

return final;

}

}

class Program

{

static void Main(string[] args)

{

String x,y;

int k;

x = Console.ReadLine();

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

y = userprogramcode.getString(x,k);

Console.WriteLine(y);

}

}

}

**123. Duplicate Date Elements**    
 Write a program to eliminate duplicate date elements in an input String Array. Print the resultant String array/list in dd/MM/yyyy format. Business Rule: 1. Input Date will be of the form: dd-MM-yyyy or dd/MM/yyyy or dd.MM.yyyy or dd-Month-yyyy. 2. If any date is invalid, print 'Invalid Date'. Create a class named UserProgramCode that has the following static method   
 public static List<string>  removeDuplicateDate(string[] input1) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
    
 Input and Output Format:  
 The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.  
 The next 'n' lines of input consist of elements in the input array.  
 Refer business rules and sample output for formatting specifications. Sample Input 1 : 5 20/09/2014 30-03-2015 13.06.2012 20-09-2014 20-September-2014    
 Sample Output 1 : 20/09/2014 30/03/2015 13/06/2012 Sample Input 2 :  
 5 20/09/2014 30-03-2015 13.13.2012 20-09-2014 20-September-2014    
 Sample Output 2 : Invalid Date

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace duplicatedate

{

class Program

{

static void Main(string[] args)

{

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

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

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

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

d.Add(Console.ReadLine());

e = pp.fun(d);

foreach (string f in e)

Console.WriteLine(f);

Console.ReadLine();

}

}

class pp

{

public static List<string> fun(List<string> d)

{

DateTime dt1;

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

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

foreach (string a in d)

{

bool i = DateTime.TryParseExact(a, "yyyy-MM-dd", null, System.Globalization.DateTimeStyles.None, out dt1);

if (i == true)

{

string k = dt1.ToString("yyyy-MM-dd");

f.Add(k);

}

if (i == false)

{

z.Add("-1");

return z;

}

}

f= f.Distinct().ToList();

return f;

}

}

}

**124.CheckCharacters** Given a method with a string input, write code to test if first and last characters are same. Incase they are same return 1 else return -1 as output. Note - Consider case.   
    
 Example:   
 Input = ""the picture was great""   
 first character = 't'   
 last character = 't'   
 Output = 1  
    
 Include a class UserProgramCode with static method checkCharacters that accepts a string and returns an integer. Create a class Program which would get the input and call the static method checkCharacters present in the UserProgramCode.    
 Input and Output Format: Input is a String - a sentence Output is a String --- “The characters are same” or “The characters are different”. Sample Input 1:  
 the picture was great  
    
 Sample Output 1:  
 The characters are same  
    
 Sample Input 2:  
 Hai how are you?  
    
 Sample Output 2:  
 The characters are different

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace program32

{

class Program

{

static void Main(string[] args)

{

int n;

n=UserProgramCode.checkcharacters("the picture was great");

Console.WriteLine(n);

}

}

class UserProgramCode

{

public static int checkcharacters(string str)

{

int len = str.Length;

string str1 = str.Substring(0, 1);

string str2 = str.Substring(len-1);

if (str1.Equals(str2))

{

return 1;

}

else

{

return -1;

}

}

}

}

**125.Boundary Average**    
 Given an int array as input, write a program to compute the average of the maximum and minimum element in the array.  
    
 Include a class UserProgramCode with a static method “getBoundaryAverage” that accepts an integer array as argument and return a avegare of max and min value.  
    
 Create a class Program which would get the input array and call the static method getBoundaryAverage present in the UserProgramCode.  
    
 Input and Output Format:  
 The first line of the input consists of an integer n, that corresponds to the size of the array.  
 The next n lines consist of integers that correspond to the elements in the array. Assume that the maximum number of elements in the array is 10.  
    
 Output consists of a single float value that corresponds to the average of the max and min element in the array. Output is displayed correct to 1 decimal place.    
    
 Sample Input :  
 6  
 3  
 6  
 9  
 4  
 2  
 5  
    
 Sample Output:  
 5.5

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication24

{

class Program

{

static void Main(string[] args)

{

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

int[] s =new int[n];

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

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

double a = UserProgramCode.getBoundaryAverage(s);

Console.WriteLine(a);

Console.ReadLine();

}

}

class UserProgramCode

{

public static double getBoundaryAverage(int[] a)

{

double d,e;

d=((a.Max()+a.Min())/2.0);

e = Math.Round(d, 1);

return e;

}

}

}

**126.Triplets**

Given inputs ,integer array input and an integer value 'k'. Write a program to find the three elements in the array whose sum is equal to the given input value 'k' and store the output in an array. Store the output elements in the same order as present in input array. Business rule: 1) If any of the inputs is negative, then print -1. 2) If the input array does not contains the triplets whose sum is equal to integer k, then print -2. 3) If the input array contains any duplicates characters, then print -3. Create a class named UserProgramCode that has the following static method   
 public static int[] findTriplets(int[] input1,int k)  
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.  
 The next 'n' lines of input consist of elements in the input array. The next line of the input corresponds to the value of k.    
 Refer business rules and sample output for formatting specifications.  
    
 Sample Input 1 : 6 12 3 4 1 6 9 24    
 Sample Output 1 : 12 3 9 Sample Input 2 : 6 12 3 4 -1 6 9 16    
 Sample Output 2 : -1

using System;

class UserProgramCode

{

public static int[] findTriplets(int[] input,int num)

{

int[] triplet=new int[3];

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

{

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

{

for (int k = j+1; k < input.Length; k++)

{

if (input[i] < 0 || input[j] < 0 || input[k] < 0)

{

triplet[0] = -1;

}

else if (input[i] == input[j] || input[j] == input[k] || input[i] == input[k])

{

triplet[0] = -3;

}

else if (input[i] + input[j] + input[k] == num)

{

triplet[0] = input[i];

triplet[1] = input[j];

triplet[2] = input[k];

}

}

}

}

if (triplet[0] + triplet[1] + triplet[2] == num)

{

return triplet;

}

else

{

triplet[0] = -2;

}

return triplet;

}

}

using System;

class Program

{

public static void Main( string[] args )

{

int size=0,i,num;

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

int[] arr = new int[size];

int[] output = new int[3];

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

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

}

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

output = UserProgramCode.findTriplets(arr,num);

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

{

if(output[i]!=0)

Console.WriteLine(output[i]);

}

Console.Read();

}

}

}

**127. Validating the pan** Write a code  to validate the given PAN number as per the following rules:  
 1. There must be eight characters.  
 2. First three letters must be alphabets followed by four digit number and ends with alphabet  
 3. All alphabets should be in uppercase.    
 Include a class UserProgramCode with static method validatePAN that accepts the String and return type should be interger.  
 Create a class Program which would get the input and call the static method validatePAN present in the UserProgramCode.  
    
 In validatePAN() if PAN card number is Valid return 1 otherwise return 2 . In Program  
 If the method returns 1 then print "Valid PAN code" If the method returns 2 then print "Invalid PAN code" Sample Input 1 ALD3245E Sample Output 1 Valid PAN code Sample Input 2 4567123 Sample Output 2

Invalid PAN code

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

namespace Validate\_pan

{

class UserPRogramCode

{

public static int ValidatePan(string input1)

{

int output1 = 0;

Regex ex = new Regex("^([A-Z]{3}[0-9]{4}[A-Z]{1})$");

if (ex.IsMatch(input1))

{

output1 = 1;

}

else

{

output1 = 2;

}

return output1;

}

}

}

class Program

{

static void Main(string[] args)

{

string str = Console.ReadLine();

int op = UserPRogramCode.ValidatePan(str);

if (op == 1)

Console.WriteLine("Valid PanCode");

else

Console.WriteLine("Invalid");

Console.ReadLine();

}

}

}

**128. Count Characters**

Write a program to count the number of characters present in the given input String array. Include a class UserProgramCode with static method countCharacters which accepts string array. The return type is a integer value which is the count of characters in the string array. Create a class Program which would get the input and call the static method countCharacters present in the UserProgramCode. Input string must contains only the alphabets then return count of characters else return the -1. If count value is -1 then print "Invalid Input". Input and Output Format : Input consists of a integer and String array. Integer represents a size of the array following by the string elements. Output consists of a integer which is the count of the character from string array. Sample Input 1: 3 cherry apple blueberry Sample Output 1: 20 Sample Input 2: 2 @aaa bbb Sample Output 2: Invalid Input

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication18

{

class UserProgramCode

{

public static int countcharachters(string[] s)

{

int sum = 0,flag = 0 ;

foreach (string s1 in s)

{

char[] ch = s1.ToCharArray();

foreach (char c in ch)

{

if (char.IsLetter(c))

{

flag++;

}

}

}

foreach (string s1 in s)

{

sum = sum + s1.Length;

}

if(flag==sum)

return sum;

else

return -1;

}

}

}

class Program

{

static void Main(string[] args)

{

UserProgramCode u=new UserProgramCode();

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

string[] s=new string[n];

int result;

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

s[i] = Console.ReadLine();

result=UserProgramCode.countcharachters(s);

if(result==-1)

Console.WriteLine("Invalid Input");

else

Console.WriteLine(result);

}

}

}

}

**129. String Occurences**  
    
 Write a program to count the number of occurences of second word of second sentence in the first sentence.  
 Return the count as output. Note - Consider case. Include a class UserProgramCode with a static method countNoOfWords which accepts two string variables. The return type is the integer.  
 Create a Class Program which would be used to accept two Input strings and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of two strings with maximum size of 100 characters.   
 Output consists of an integer.  
 Refer sample output for formatting specifications.  
 Sample Input 1:  
 abc bcd abc bcd abc abc  
 av abc  
 Sample Output 1:  
 4  
 Sample Input 2:  
 ABC xyz AAA  
 w abc  
 Sample Output 2:  
 0

using System;

class UserProgramCode

{

public static int countNoOfWords(string str1,string str2)

{

string[] sub1=str1.Split(' ');

string[] sub2 = str2.Split(' ');

int ctr=0;

string str = sub2[1];

for (int i = 0; i < sub1.Length; i++)

{

if (sub1[i] == str)

{

ctr++;

}

}

return ctr;

}

}

class Program

{

public static void Main( string[] args )

{

string str1=Console.ReadLine();

string str2=Console.ReadLine();

Console.WriteLine(UserProgramCode.countNoOfWords(str1,str2));

Console.ReadLine();

}

}

**130. Find Common Elements**  
 Write a program to find the common elements from the 2 integer lists and to print them in ascending order.  
    
 Example:   
 input1: [4,7,3,9,1,5]  
 input2: [10,4,6,5,3]  
    
 Output1:[3,4,5]   
    
 Business Rules:  
    
 Only positive numbers should be given to the input Lists.  
    
 1. If the input1 List consists of negative numbers, store  -1 in the list returned from the method..  
 2. If the input2 List consists of negative numbers, store -2 in the list returned from the method.  
 3. If both the input lists, input1 and input2  consists of negative numbers,store -3 in the list returned from the method.  
    
 Include a class UserProgramCode with a static method FindCommonElements which accepts two integers (size of the two integer list), and the two integer lists. The return type (integer list) should return the output integer list. Store -1,-2 or -3 in the output list according to the business rules and return if necessary.  
 Create a Class Program which would be used to accept two integers (size of the two integer list), and the two integer lists, and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of two integers (size of the two integer list), and the two integer lists.  
 Output consists of an Integer list or, a String “First list is negative” if  -1 is returned, “ Second list is negative” if -2 is returned, or “Both lists are negative” if -3 is returned.  
    
 Refer sample output for formatting specifications.  
    
 Sample Input 1:  
 6  
 5  
 4  
 7  
 3  
 9  
 1  
 5  
 10  
 4  
 6  
 5  
 3  
 Sample Output1:  
 3  
 4  
 5  
    
 Sample Input 2:  
 6  
 5  
 -4  
 7  
 3  
 9  
 1  
 5  
 10  
 4  
 6  
 5  
 3  
 Sample Output2:  
 First list is negative  
    
    
 Sample Input 3:  
 6  
 5  
 4  
 7  
 3  
 9  
 1  
 5  
 10  
 4  
 6  
 5  
 -3  
 Sample Output 3:  
 Second list is negative  
    
 Sample Input 4:  
 6  
 5  
 -4  
 7  
 3  
 9  
 1  
 5  
 10  
 4  
 6  
 5  
 -3  
 Sample Output 4:  
 Both lists are negative

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace fFindCommon

{

class UserProgramCode

{

public static List<int> FindCommon(List<int> list1, List<int> list2)

{

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

int flag = 0;

int flag1 = 0;

for (int i = 0; i < list1.Count; i++)

{

if (list1[i] < 0)

{

final.Add(-1);

flag = 1;

}

}

for (int i = 0; i < list2.Count; i++)

{

if (list2[i] < 0)

{

final.Add(-2);

flag1 = 2;

}

}

if (flag == 1 && flag1 == 2)

{

final.Clear();

final.Add(-3);

return final;

}

else if (flag == 1)

return final;

else if(flag1==2)

return final;

else

for (int i = 0; i < list1.Count; i++)

{

for (int j = 0; j < list2.Count; j++)

{

if(list1[i]==list2[j])

{

final.Add(list1[i]);

}

}

}

final.Sort();

return final;

}

}

}

class Program

{

static void Main(string[] args)

{

int n1 = int.Parse(Console.ReadLine());

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

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

{

list.Add(Convert.ToInt32(Console.ReadLine()));

}

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

List<int> list2 = new List<int>();

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

{

list2.Add(Convert.ToInt32(Console.ReadLine()));

}

List<int> op = UserProgramCode.FindCommon(list, list2);

foreach (int item in op)

{

Console.WriteLine(item);

}

Console.ReadLine();

}

}

}

**131. Longest Word**    
 Write a Program which finds the longest word from a sentence. Your program should read a sentence as input from user and return the longest word. In case there are two words of maximum length return the word which comes first in the sentence.  
 Include a class UserProgramCode with a static method getLargestWord which accepts a string. The return type is a string that corresponds to the largest word in the sentence.  
 Create a Class Program which would be used to accept a input string and call the static method present in UserProgramCode.    
 Input and Output Format:  
 Input consists of a string with a maximum size of 100 characters.   
 Output consists of a single string.  
 Refer sample output for formatting specifications.  
 Sample Input 1:  
 Welcome to the world of Programming  
 Sample Output 1:  
 Programming  
 Sample Input 2:  
 ABC DEF  
 Sample Output 2:  
 ABC

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace level1\_57

{

class UserProgramCode

{

public static string getLargestWord(string str)

{

int l=0,max=0,ind=-1;

string[] s=new string[100];

s = str.Split(' ');

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

{

l = s[i].Length;

if (max < l)

{

max = l;

ind = i;

}

}

return s[ind];

}

}

class Program

{

static void Main(string[] args)

{

string str,str1;

str = Console.ReadLine();

str1 = UserProgramCode.getLargestWord(str);

Console.WriteLine(str1);

Console.Read();

}

}

}

**132. Largest Digit** Write a program to find the Largest digit from given input integer. Print the largest digit. If the number is negative, print “Negative Number”.  
    
 Example:   
 Input1: 524  
 Output1: 5  
    
 Include a class UserProgramCode with a static method findLargestDigit which accepts an integer. The return type (integer) should return the largest digit. Return -1 if the number is negative.  
 Create a Class Program which would be used to accept an integer and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of an integer.  
 Output consists of an integer (the largest digit), or a String “ Negative Number” if the input is negative.  
 Refer sample output for formatting specifications.  
    
 Sample Input 1:  
 524  
 Sample Output 1:  
 5  
    
 Sample Input 2:  
 -23  
 Sample Output 2:  
 Negative Number

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication100

{

class Program

{

static void Main(string[] args)

{

int a;

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

int c = UserProgramCode.findLargestDigit(a);

if(c==-1)

Console.WriteLine("Negative Number");

else

Console.WriteLine(c);

}

}

class UserProgramCode

{

public static int findLargestDigit(int b)

{

if (b >= 0)

{

string a = b.ToString();

int l = a.Length;

int c = 0;

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

{

int x = b / 10;

int y = b % 10;

b = x;

if (y >= c)

{

c = y;

}

}

return c;

}

else

return (-1);

}

}

}

**133. Dash Check** Write a program to read two strings and check whether or not they have dashes in the same places. Print “Yes” if the condition satisfies, else print “No”.  
 Include a class UserProgramCode with a static method compareDashes which accepts two strings and returns an integer. The function returns 1 if all dashes are placed correctly, else the function returns 2.  
 Create a Class Program which would be used to accept two strings and call the static method present in UserProgramCode.  
 Note: The strings must have exactly the same number of dashes in exactly the same positions. The strings might be of different length.  
 Input and Output Format:  
 Input consists of two strings.  
 Output consists of a string (“Yes” or “No”).  
 Refer sample output for formatting specifications.  
 Sample Input 1:  
 hi—there-you.  
 12--(134)-7539  
 Sample Output 1:  
 Yes  
    
 Sample Input 2:   
 -15-389  
 -xyw-zzy  
 Sample Output 2:  
 No

class Program

{

public static void Main(string[] args)

{

string input1, input2;

input1 = Console.ReadLine();

input2 = Console.ReadLine();

int value = UserProgramCode.compareDashes(input1, input2);

if (value == 1)

Console.WriteLine("Yes");

else

Console.WriteLine("No");

Console.ReadLine();

}

}

class UserProgramCode

{

public static int compareDashes(string input1, string input2)

{

// fill your code here

char[] s1 = input1.ToCharArray();

char[] st1 = input2.ToCharArray();

int len, count = 0;

if (s1.Length > st1.Length)

{

len = s1.Length;

}

else

{

len = st1.Length;

}

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

{

if (s1[i] == '-')

{

if (st1[i] != '-')

{

count++;

break;

}

}

}

if (count > 0)

return 2;

else

return 1;

**134. Arrange After Cubing** Write a code to insert cube of a number in between two numbers in an integer array if those numbers satisfy the below conditions: Conditions: 1) The elements in the array must be consecutive numbers 2) Second element in the array should be the square of first element in the array. 3) Cube of first number must be inserted in between the first element and second element. 4) If any of the element in the array does not satisy the above condition add the element in the output array and proceed with the next element. Business Rules: 1. If no consecutive numbers are present assign all the elements in the Input array to the Output array . 2. If input array consists of negative elements perform the square of negative numbers and then print the final output array. Create a class named UserProgramCode that has the following static method   
 public static int[] arrangeAfterCubing(int[] input1)  
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.  
 The next 'n' lines of input consist of elements in the input array.  
 Refer sample output for formatting specifications.  
    
 Sample Input 1: 7 1 2 4 6 7 3 9 Sample Output 1: 1 2 8 4 6 7 3 27 9 Sample Input 2:  
 3 1 -4 5  
 Sample Output 2: 1 16 5

using System;

class Program

{

public static void Main( string[] args )

{

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

int[] value=new int[n];

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

{

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

}

int[] output = UserProgramCode.arrangeAfterCubing(value);

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

{

Console.Write(output[i]+"\n");

}

}

}

using System;

using System.Text;

using System.Collections;

class UserProgramCode

{

public static int[] arrangeAfterCubing(int[] input1)

{

int i=0,ch,c;

int ch1=0,flag=1;

int s = input1.Length;

ArrayList sb = new ArrayList();

int[] op1 = new int[s];

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

{

if (input1[i] < 0)

{

flag = 0;

ch1 = input1[i] \* input1[i];

input1[i] = ch1;

}

}

if (flag == 0)

{

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

{

op1[i] = input1[i]; ;

}

return op1;

}

else

{

for (i = 0; i < s - 1; i++)

{

ch = input1[i] \* input1[i];

if (input1[i + 1] == ch)

{

c = input1[i] \* input1[i] \* input1[i];

if (i == 0)

{

sb.Add(input1[i]);

sb.Add(c);

sb.Add(input1[i + 1]);

}

else

{

sb.Add(c);

sb.Add(input1[i + 1]);

}

}

else

{

if (i == 0)

{

sb.Add(input1[i]);

sb.Add(input1[i + 1]);

}

else

{

sb.Add(input1[i + 1]);

}

}

}

int len = sb.Count;

int[] op = new int[len];

i = 0;

foreach (int n in sb)

{

op[i] = n;

i++;

}

return op;

}

}

}  
 **135. String Equal Check**    
 Given two strings Input1 and Input2 and integer Input3, write a program to check if Nth character of Input1 traversing from first and Nth character of Input2 traversing from last are same irrespective of case where N is the Input3 value. Ignore case. If both are same, then print "The character is x" where x is the Nth character If both are not same, then print "The character x and y does not match" where x is the Nth character of Input1 starting from first and y is the Nth character of Input2 starting from last. Business rule: 1) If the Input1 string contains any special characters or numbers, then print 'Invalid Input' 2) If the Input2 string contains any special characters or numbers, then print 'Invalid Input' 3) If the Input3 value is greater than the length of Input1 and/or Input2, then print 'Invalid Input' Create a class named UserProgramCode that has the following static method   
 public static string stringEqualCheck(string input1, string input2, int input3)  
    
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:  
 The first line of the input consists of a string that corresponds to input1.  
 The second line of the input consists of a string that corresponds to input2.  
 The third line of the input consists of an integer that corresponds to input 3.  
 Output consists of a string. Refer business rules and sample output for the format.  
    
 Sample Input 1: Battle Final 2 Sample Output 1:   
 The character is a Sample Input 2 : Photograph Anticipate 4  
 Sample Output 2:  
 The character t and p does not match Sample Input 3 : xerox pretty 15 Sample Output 3 :  
 Invalid Input

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Collections;

namespace ConsoleApplication23

{

class UserProgramCode

{

public static string stringEqualCheck(string input1, string input2, int input3)

{

string i1, i2;

int i3;

i1 = input1;

char[] i11 = i1.ToCharArray();

i2 = input2;

char[] i22 = i2.ToCharArray();

i3 = input3;

foreach (char ch in i11)

{

if (char.IsDigit(ch))

{

Console.WriteLine("invalid input");

}

if (!char.IsLetter(ch))

{

Console.WriteLine("invalid input");

}

}

foreach (char ch in i22)

{

if (char.IsDigit(ch))

{

return "invalid input";

}

if (!char.IsLetter(ch))

{

return "invalid input";

}

}

if (i3 > i1.Length)

{

return "invalid input";

}

else if (i3 > i2.Length)

{

return "invalid input";

}

else if ((i11[i3 - 1]) == (i22[(i2.Length) - i3]))

{

return "the character is " + i11[i3 - 1];

}

else

{

return "the character " + i11[i3 - 1] + " and " + i22[(i2.Length) - i3] + " does not match";

}

}

}

}

###########################

using System;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using TestPractice;

using ConsoleApplication23;

//using ConsoleApplication9;

namespace TestPractice

{

class Program

{

static void Main(string[] args)

{

string s1, s2;

int index;

s1 = Console.ReadLine();

s2 = Console.ReadLine();

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

string disp;

disp = UserProgramCode.stringEqualCheck(s1, s2, index);

//if(disp=="1")

// Console.WriteLine("the character is " + i11[i3 - 1]);

Console.WriteLine(disp);

Console.ReadLine();

}

}

}

136. Calculate VAT

ABC stores needs a computerized solution for calculating the VAT for the billing amount. Write a code to calculate the VAT amount for the input bill amount. The VAT should be calculated in the following basis,   Type -------  VATPercentage,   Medical (M) - - - - - - - - - -- 9%   Vegetables (V) and fruits - - - - - - - - - - - 5%   Clothes (C) - - - - - - - - - - - - 12%   Electronics (E) - - - - - - - - - - -- - - 6.25%   Business Rules : 1. The codes 'M','V','C' or 'E' only should be given as input for indicating the Medical, Vegetables and Fruits, Clothes and Electronics type of goods respectively. Any other character other than the above is given as input, it is Invalid Input. 2. Only Positive number should be given as a input for bill amount. Else it is Invalid Input. Include a class UserProgramCode with a static method calculateVAT which accepts a character and double and returns a double which corresponds to the calculated VAT amount.  If the input is invalid, the method returns -1.  
    
 Create a class Program  which would get the input and call the static method calculateVAT present in the UserProgramCode. If the method returns -1, print 'Invalid Input'. Input and Output Format: Input consists of character and double. Character denotes a goods type and double denotes total amount. Refer sample output for formatting specifications. Sample Input 1 : M 70 Sample Output 1 : 6.3 Sample Input 2 : V -500 Sample Output 2 : Invalid Input

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace final

{

class UserProgramCode

{

public static double CalculateVat(char c, double input1)

{

double percent=0;

double vat=0;

double output1=0;

if (input1 < 0)

return -1;

switch (c)

{

case 'M':

percent = 9;

vat = percent / 100;

output1 = input1\* ( vat);

return output1;

case 'V':

percent = 5;

vat = percent / 100;

output1 = input1 \* ( vat);

return output1;

case 'C':

percent = 12;

vat = percent / 100;

output1 = input1 \* ( vat);

return output1;

case 'E':

percent = 6.25;

vat = percent / 100;

output1 = input1 \* ( vat);

return output1;

default:

output1 = -1;

break;

}

return output1;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace final

{

class Program

{

static void Main(string[] args)

{

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

double input1 = Convert.ToDouble(Console.ReadLine());

double op = UserProgramCode.CalculateVat(c, input1);

if (op == -1)

Console.WriteLine("Invalid Input");

else

Console.WriteLine(op);

Console.ReadLine();

}

}

}

**137. Get word with Maximum Vowels** Write a method that accepts a string input and returns the word with maximum number of vowels. In case there are two or more words with maximum number of vowels, return the first word.  
    
 Example:  
 Input: Appreciation is the best way to motivate  
 Output: Appreciation (total vowels = 6)  
    
 Include a class UserProgramCode with a static method getWordWithMaximumVowels that accepts a string and returns a string.  
    
 Create a class Program which would get the input and call the static method getWordWithMaximumVowels present in the UserProgramCode.  
    
 Input and Output Format:  
 Input consists of a string.  
 Output consists of a string. Sample Input1 : Appreciation is the best way to motivate Sample Output1 : Appreciation Sample Input2 : Sun rises in the east Sample Output2 : rises  
  

using System.Linq;

using System.Text;

namespace get\_the\_word\_with\_max\_vowles

{

class Program

{

static void Main(string[] args)

{

string str = Console.ReadLine();

string op = UserProgramcode.GETMAXVOWELS(str);

Console.WriteLine(op);

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace get\_the\_word\_with\_max\_vowles

{

class UserProgramcode

{

public static string GETMAXVOWELS(string str)

{

string[] arr = str.Split(' ');

int max = 0, lenmax = 0, count;

foreach (string element in arr)

{

char[] ch = element.ToCharArray();

count = 0;

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

{

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

count++;

}

if (count > max)

{

max = count;

lenmax = ch.Length;

}

}

string output = string.Empty;

foreach (string item in arr)

{

if (item.Length == lenmax)

{

output = item;

break;

}

}

return output;

}

}

}

**138. Sort the list** Write a program which reads an Integer(size of the list), a String List and a character, and to get the strings that will not start with the given character irrespective of case. Sort the elements in ascending order based on its length. Print the output list. If the elements are having the same length, then display the elements in alphabetical order.  
 Include a class UserProgramCode with static method  GetTheElements which accepts a String list  and a character. The return type is List<String>.  
 Create a Class Program which would be used to get the inputs and call the static method present in UserProgramCode. In GetTheElements method  
 Only alphabets should be given in list , otherwise return "Invalid Input". When the output list is empty, then return "List is Empty". Otherwise return the appropriate result.  
 In Program class Print the result which is return by GetTheElements method in UserProgramCode. Input output format The first line of the input is an integer that corresponds to n, the size of the list. The next n lines of input correspond to the elements in the string list. The line of the input contains the  character. The output is the List type List<String>. Sample Input 1: 3 read write edit e Sample Output 1: read write Sample Input2 : 2 Elegent event e Sample Output2 : List is Empty Sample Input 3: 2 Eleg$ent e^ent e Sample Output 3 : Invalid Input

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace progm55

{

class UserProgramCode

{

public static int[] sortList(int[] a)

{

int temp;

int maxi = a.Length;

if (maxi == 0)

{

a[0] = -1;

}

else

{

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

{

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

{

if (a[i] < 0)

{

a[0] = -1;

}

else

{

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

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

}

}

return a;

}

}

class Program

{

static void Main(string[] args)

{

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

int[] a=new int[n];

int[] output=new int[n];

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

{

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

}

output=UserProgramCode.sortList(a);

if (output[0] != -1)

{

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

{

Console.WriteLine(output[i]);

}

}

else

{

Console.WriteLine("Invalid Input");

}

}

}

}

139. Matching String  
 Write a program to display the strings which starts with the character passed in input2 variable from input1 list irrespective of case. Store the result in output list in the same sequence in which they are found in input1 list. Then form a string in output1 list as in the example given below. In the example given below, 2 corresponds to the numer of strings in the input list that start with the given character.  
    
 Example:  
    
 input1: [abc,apple,Mango]  
 input2: a  
    
 Output1:[abc\_2,apple\_2]  
    
 Business Rule:  
    
 1. If there is no match found in input1 list then store "-1" in the string list returned from from the method and print “No match found” in Main.  
    
 Include a class UserProgramCode with a static method sortStrings which accepts an integer (the size of the string list), a String list and a character. The return type (String List) should return the output String List.  
 Create a Class Program which would be used to accept an integer, a String List and a character , and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of an integer, a String list and a character, where the integer corresponds to the size of the List, String list corresponds to the input string list and the character values corresponds to the starting character.  
 Output consists of a String List.  
    
 Refer sample output for formatting specifications.  
    
 Sample Input 1:  
 3  
 abc  
 apple  
 Mango  
 a  
 Sample Output 1:  
 abc\_2  
 apple\_2  
    
 Sample Input 2:  
 3  
 abc  
 apple  
 Mango  
 b  
 Sample Output 2:  
 No match found

using System;

using System.Collections.Generic;

class Program

{

public static void Main(string[] args)

{

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

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

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

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

{

list.Add(Console.ReadLine());

}

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

result = UserProgramCode.SortStrings(size, list, ch);

if (result[0].Equals("-1"))

Console.WriteLine("No match found");

else

{

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

Console.WriteLine(result[i]);

}

}

}

using System;

using System.Collections.Generic;

class UserProgramCode

{

public static List<string> SortStrings(int size, List<string> li, char ch)

{

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

// int size = Convert.ToInt32(Console.ReadLine());

int k = 0, count = 0;

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

{

char c = Convert.ToChar(li[i].Substring(0, 1));

if (ch == c)

count++;

}

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

{

string item;

char c = Convert.ToChar(li[i].Substring(0, 1));

if (ch == c)

{

item = li[i] + "\_" + count;

result.Add(item);

k++;

}

}

return result;

}

}

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

Matching string(SortStrings)

using System;

using System.Collections.Generic;

class Program

{

public static void Main(string[] args)

{

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

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

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

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

{

list.Add(Console.ReadLine());

}

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

result = UserProgramCode.SortStrings(size, list, ch);

if (result[0].Equals("-1"))

Console.WriteLine("No match found");

else

{

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

Console.WriteLine(result[i]);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

class UserProgramCode

{

public static List<string> SortStrings(int size, List<string> li, char ch)

{

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

int c = 0;

foreach (string item in li)

{

if (item.Contains(ch))

{

c++;

}

}

c = c - 1;

li.RemoveAt(size - 1);

if (c == 0 )

{

res.Add("-1");

return res;

}

else

{

foreach (var item in li)

{

if (item.Contains(ch))

{

res.Add(item + "\_" + c);

}

}

}

return res;

}

}

**140. Count Sequential Characters** Get a string as input and write code to count the number of characters which gets repeated 3 times consecutively and return that count (ignore case).   
    
 Include a class UserProgramCode with a static method countSequentialChars which accepts a string as input and return type is an integer.  
 The method returns the repeat count. If no character gets repeated 3 times consecutively the method returns -1.  
    
 Create a class Program which would get the input and call the static method countSequentialChars present in the UserProgramCode. If the method returns -1, print 'No Repeated Words Found'.  
    
    
 Input and Output Format:  
 Input consists a string.  
 Refer sample output for formatting specifications.  
    
 Sample Input 1:  
 abcXXXabc  
 Sample Output 1:  
 1  
 Sample Input 2:  
 aaxxyzAAx  
 Sample Output 2:  
 No Repeated Words Found

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication24

{

class Program

{

static void Main(string[] args)

{

string s = Console.ReadLine();

int a = UserProgramCode.countSequentialChars(s);

if(a==-1)

Console.WriteLine("No Repeated Words Found");

else

Console.WriteLine(a);

Console.ReadLine();

}

}

class UserProgramCode

{

public static int countSequentialChars(string s)

{

int l = s.Length;

string[] st = new string[50];

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

{

st[k] = s.Substring(k, 1);

}

int count = 0;

int c=0;

for (int k = 0; k < l - 1; k++)

{

if (st[k] == st[k+1])

count++;

else

count = 0;

if (count == 2)

c++;

}

if(c==0)

return -1;

else

return c;

}

}

}

**141. Check Supply** The Policy followed by a Company to process Customer Orders is given by the following rules:  
 Rules:  
 (a) If a Customer Order is less than or equal to that in Stock and if the Credit is OK, Supply the required quantity.  
 (b) If the Credit is Not OK, then do not Supply. Send him intimation saying "Cannot Supply".  
 (c) If the Credit is OK, but the item in Stock is less than the order, Supply what is in Stock. Intimate to him that the balance will be shipped.  
 (c) If the Credit is OK and the item in Stock is 0, Output should be "Out Of Stock".  
    
 Input1- Stock in hand  
 Input2- Customer Order Quantity  
 Input3- Credit (true -OK, false -Not OK)  
    
 Output1- Message(“Supply","Cannot Supply","Out Of Stock","Balance Will Be Shipped Later")  
    
 Example:   
    
 Input1: 50  
 Input2: 5  
 Input3: true  
    
 Output1: Supply  
    
 Include a class UserProgramCode with a static method checkSupply which accepts two Integers and a Boolean value. The return type (String) should return a String according to the business rules.  
 Create a Class Program which would be used to accept two Integers and a Boolean value, and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of a 2 integers and a boolean value.  
 Output consists of a String.  
    
 Refer sample output for formatting specifications.  
    
 Sample Input 1:  
 50  
 5  
 true  
 Sample Output 1:  
 Supply  
    
 Sample Input 2:  
 50  
 5  
 false  
 Sample Output 2:  
 Cannot Supply  
    
 Sample Input 3:  
 50  
 55  
 true  
 Sample Output 3:  
 Balance Will Be Shipped Later  
    
 Sample Input 4:  
   
 5  
 true  
 Sample Output 4:  
 Out Of Stock

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication18

{

class UserProgramCode

{

public static string checksupply(int n1, int n2,bool value)

{

if (value == true && n1 == 0)

{

return ("OutOfStock");

}

else

if (value == true && n2 < n1 | n1 == n2)

{

return ("Supply");

}

else

if (value == true && n1 < n2)

{

return ("Balance Will Be Shipped Later");

}

else

if (value == false)

{

return "Cannot Supply";

}

else

return "";

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication18

{

class Program

{

static void Main(string[] args)

{

UserProgramCode u = new UserProgramCode();

int n1, n2;

bool value;

string output;

n1 = int.Parse(Console.ReadLine());

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

value=bool.Parse(Console.ReadLine());

output = UserProgramCode.checksupply(n1, n2, value);

Console.WriteLine(output);

}

}

}

**142. Reverse Substring**    
 Given a input string with a startIndex and length, Write a program to extract substring from right to left. Assume the last character has index 0.  
 Include a class UserProgramCode with a static method reverseSubstring which accepts a string and two integers. The return type is string as given in the above statement.  
 Create a Class Program which would be used to accept Input and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of a string, and two integers – startIndex and length.   
 Output consists of a string as mentioned in the problem statement.  
 Refer sample output for formatting specifications.  
 Sample Input 1:  
 rajasthan  
 2  
 3  
 Sample Output 1:  
 hts

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication9

{

public class UserProgramCode

{

public static string reverseSubstring(string str, int start, int len)

{

//StringBuilder sb = new StringBuilder();

char[] ch = str.ToCharArray();

Array.Reverse(ch);

string s=new string(ch);

string outp = s.Substring(start, len);

//foreach (char item in ch)

//{

// sb.Append(item);

//}

return outp;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using TestPractice;

using ConsoleApplication9;

namespace TestPractice

{

class Program

{

static void Main(string[] args)

{

string str=Console.ReadLine();

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

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

string str2 = UserProgramCode.reverseSubstring(str, start, len);

Console.WriteLine(str2);

Console.ReadLine();

}

}

}

**143. Add Years**

Write a method which can display the date n years after the given date. The date should be given in string format "mm/dd/yyyy" without time and the resultant added date should also be in the format "mm/dd/yyyy" . “n” as Years and the given date should be accepted as an argument. Include a class UserProgramCode with a static method addYears which accepts a string as input and output should be a string.  
   Business Rules : 1) Only positive value should be given as input to the integer else the method returns -2. 2) If the date format is not "mm/dd/yyyy" given in the string, the method returns -1. 3) Otherwise return the corresponding date in given format. Create a class Program which would get the input and call the static method addYears present in the UserProgramCode. If the method returns -1, print 'Invalid date format'. If the method returns -2, print 'Invalid Input'. Input and Output Format : Input consists of a string which accepts a date. Refer the sample input and output for formatting specifications. Sample Input 1 : 10/21/2009 5 Sample Output 1 : 10/21/2014 Sample Input 2 : 2009/10/03 -2 Sample Output 2 : Invalid Input Sample Input 3 : 27-10-2009 1 Sample Output 3 : Invalid date format

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Globalization;

namespace AddYears

{

class Program

{

public static void Main(string[] args)

{

string inputDate = Console.ReadLine();

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

string output = UserProgramCode.addYears(inputDate, day);

if (output.Equals("-1"))

Console.WriteLine("Invalid Input");

else if (output.Equals("-2"))

Console.WriteLine("Invalid date format");

else

Console.WriteLine(output);

Console.ReadLine();

}

}

class UserProgramCode

{

public static string addYears(string date, int day)

{

// fill your code here

DateTime dt;

string output;

if (day < 0)

{

return "-1";

}

else

{

bool res = DateTime.TryParseExact(date, "MM/dd/yyyy", null, System.Globalization.DateTimeStyles.None, out dt);

if (res == true)

{

dt = dt.AddDays(day);

output = dt.ToString("MM/dd/yyyy");

return output;

}

else

{

return "-2";

}

}

}

}

}

**144. Count Characters**  
 Given a string array as input, write a program to find the total number of characters in all the words (in the given string array). Assume that all strings are single words.  
    
 Create a class named UserProgramCode that has the following static method  
 public static int countCharacters(string[] input1)  
 Create a class named Program that accepts a string array as input and calls the static method present in the UserProgramCode.  
 Input and Output Format:  
 The first line of the input consists of an integer 'n' that corresponds to the number of elements in the string array.  
 The next 'n' lines of the input consists of strings that correspond to the elements in the string array.  
 Output consists of a single integer that corresponds to the total number of characers in all the words in the given string array.  
 Sample Input : 2 cherry apple Sample Output : 11

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication18

{

class UserProgramCode

{

public static int countcharachters(string[] s)

{

int sum = 0,flag = 0 ;

foreach (string s1 in s)

{

char[] ch = s1.ToCharArray();

foreach (char c in ch)

{

if (char.IsLetter(c))

{

flag++;

}

}

}

foreach (string s1 in s)

{

sum = sum + s1.Length;

}

if(flag==sum)

return sum;

else

return -1;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication18

{

class Program

{

static void Main(string[] args)

{

UserProgramCode u=new UserProgramCode();

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

string[] s=new string[n];

int result;

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

s[i] = Console.ReadLine();

result=UserProgramCode.countcharachters(s);

if(result==-1)

Console.WriteLine("Invalid Input");

else

Console.WriteLine(result);

}

}

}

**145. Find Gift Voucher** In a game two dice is thrown. From the sum of the two dice, the player is going to get the gift voucher from the club. Write a program to find the amount of the gift voucher. Print the amount received as gift.  
    
 Sum of Two Dices ------------------ Gift Voucher in Rs  
 2,3,6,11 ------------------------------ 1000  
 4,7,10--------------------------------- 3000  
 5,8,9,12------------------------------- 5000  
    
 In the method,  
 Only Positive number (1-6) should be given as a input numbers. Else return -1.  
    
 Include a class UserProgramCode with a static method findGiftVoucher which accepts two integers. The return type (Integer) should return the gift voucher amount. If the any of the inputs is invalid return -1.  
 Create a Class Program which would be used to accept a positive Integer, and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of two integers.  
 Output consists of an Integer( the gift voucher amount) or a String “Invalid Input” if any of the inputs is invalid.  
    
 Refer sample output for formatting specifications.  
    
 Sample Input 1:  
 1  
 2  
 Sample Output 1:  
 1000  
    
 Sample Input 2:  
 1  
 -2  
 Sample Output 2:  
 Invalid Input

using System;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using TestPractice;

namespace TestPractice

{

class Program

{

static void Main(string[] args)

{

int n, c, m;

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

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

c = userprogramcode.findGiftVoucher(n, m);

if (c == -1)

{

Console.WriteLine("invalid input");

}

else

{

Console.WriteLine(c);

}

Console.ReadLine();

}

}

}

##############

class userprogramcode

{

public static int findGiftVoucher(int a, int b)

{

if (a > 0 && b > 0 && a < 7 && b < 7)

{

if ((a + b == 2) || (a + b == 3) || (a + b == 6) || (a + b == 11))

return (1000);

else if ((a + b == 4) || (a + b == 7) || (a + b == 10))

return (3000);

else if ((a + b == 5) || (a + b == 8) || (a + b == 9) || (a + b == 12))

return (5000);

}

else return (-1);

return 0;

}

}

}

**146. Maximum Vowels**  
 Given a sentence as string input, write a program to fetch the word with maximum number of vowels and print it. In case there are two or more words with maximum number of vowels, print the first word.    
 Create a class named UserProgramCode that has the following static method   
 public static string getWordWithMaximumVowels(string input1) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 Input consists of a string.  
 Refer business rules and sample output for formatting specifications. Sample Input : Appreciation is the best way to motivate Sample Output :  
 Appreciation

using System;

class Program

{

public static void Main( string[] args )

{

string input = Console.ReadLine();

string result = UserProgramCode.getWordWithMaximumVowels(input);

Console.WriteLine(result);

Console.ReadLine();

}

}

using System;

class UserProgramCode

{

public static string getWordWithMaximumVowels(string inpt)

{

string result = "";

int maxvowels = 0;

string input=inpt.ToLower();

string[] array = input.Split();

foreach (string str in array)

{

int vowels = 0;

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

{

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

vowels++;

}

if (vowels > maxvowels)

{

maxvowels = vowels;

result = str;

}

}

return result;

}

}

**147. Travel Agency** A travel agency has set standard tariffs for their pick up - drop services in a particular route.The route covers A,B,C,D locations one after the other. A. Tariff for the travel from Location A to Location B is 10 units/Km B. Tariff for the travel from Location B to Location C is 20 units/Km C. Tariff for the travel from Location C to Location D is 40 units/Km Return journey service is also provided. The starting point, destination point and the Time of travel ( Normal - N, Untime - U) covered by a vehicle in a day are given as input1 in the format {XYZ...} - here X represents Start point , Y represents the destination point and Z represents the Time of travel. For untime travel,20% additional charges are applicable on actual tariff for that route. Write a program to calculate the total tariff collected by that vehicle for the day given and print the output in the following format, The car has taken A trips and has collected total amount of C rupees. -Here A refers to the total number of services provided per day and C refers to the total amount from all the travels. Business rules: 1.If start point or destination points are invalid (other than A,B,C,D), print 'Invalid Location'. 2.If Time of travel is not either N or U , print 'Invalid Time of Travel'.    
 Create a class named UserProgramCode that has the following static method   
 public static int getTariffAmount(string[] input1)  
    
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:    
 The first line of the input consists of an integer, n that corresponds to the number of elements in the string array.  
 The next 'n' lines of input consists of strings that correspond to elements in the string array.  
 Refer business rules and sample output for output format.  
 Always display the tariff to be paid as an int.  
    
 Sample Input 1 : 4 ACN DAU ADN DCU    
 Sample Output 1 : The car has taken 4 trips and has collected total amount of 232 rupees Sample Input 2 :  
 4 ACN FAU ADN DCU    
 Sample Output 2 : Invalid Location

using System;

using System.Text.RegularExpressions;

namespace code1

{

class Program

{

static void Main(String[] args)

{

int n, amount;

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

String[] input1=new String[n];

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

{

input1[i] = Console.ReadLine();

}

amount=UserMainCode.getTariffAmount(input1);

if(amount!=-1&& amount!=-2)

Console.WriteLine("The car has taken "+n+" trips and has collected total amount of " + amount + " rupees");

}

}

}

using System;

public class UserMainCode

{

public static int getTariffAmount(string[] input1)

{

int length = input1.Length;

double amount = 0;

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

{

if (input1[i][2] == 'N')

{

if (input1[i][0] == 'A')

{

if (input1[i][1] == 'B')

amount += 10;

else if (input1[i][1] == 'C')

amount += 30;

else if (input1[i][1] == 'D')

amount += 70;

}

else if (input1[i][0] == 'B')

{

if (input1[i][1] == 'A')

amount += 10;

else if (input1[i][1] == 'C')

amount += 20;

else if (input1[i][1] == 'D')

amount += 60;

else

{

Console.WriteLine("Invalid Location"); return -1;

}

}

else if (input1[i][0] == 'C')

{

if (input1[i][1] == 'A')

amount += 30;

else if (input1[i][1] == 'B')

amount += 20;

else if (input1[i][1] == 'D')

amount += 40;

else

{

Console.WriteLine("Invalid Location"); return -1;

}

}

else if (input1[i][0] == 'D')

{

if (input1[i][1] == 'A')

amount += 70;

else if (input1[i][1] == 'B')

amount += 60;

else if (input1[i][1] == 'C')

amount += 40;

else

{

Console.WriteLine("Invalid Location"); return -1;

}

}

else

{

Console.WriteLine("Invalid Location"); return -1;

}

}

else if (input1[i][2] == 'U')

{

if (input1[i][0] == 'A')

{

if (input1[i][1] == 'B')

amount += 10 \* 1.2;

else if (input1[i][1] == 'C')

amount += 30 \* 1.2;

else if (input1[i][1] == 'C')

amount += 70 \* 1.2;

}

else if (input1[i][0] == 'B')

{

if (input1[i][1] == 'A')

amount += 10 \* 1.2;

else if (input1[i][1] == 'C')

amount += 20 \* 1.2;

else if (input1[i][1] == 'D')

amount += 60 \* 1.2;

else

{

Console.WriteLine("Invalid Location"); return -1;

}

}

else if (input1[i][0] == 'C')

{

if (input1[i][1] == 'A')

amount += 30 \* 1.2;

else if (input1[i][1] == 'B')

amount += 20 \* 1.2;

else if (input1[i][1] == 'D')

amount += 40 \* 1.2;

else

{

Console.WriteLine("Invalid Location"); return -1;

}

}

else if (input1[i][0] == 'D')

{

if (input1[i][1] == 'A')

amount += 70 \* 1.2;

else if (input1[i][1] == 'B')

amount += 60 \* 1.2;

else if (input1[i][1] == 'C')

amount += 40 \* 1.2;

else

{

Console.WriteLine("Invalid Location"); return -1;

}

}

else

{

Console.WriteLine("Invalid Location"); return -1;

}

}

else

{ Console.WriteLine("Invalid Time of Travel"); return -2; }

}

return (int)amount;

}

}

**148. Add Days** Write a program which can print the date n days after the given date.  
 The date should be given in string format “mm/dd/yyyy” without time and the resultant added date should also be in the format “mm/dd/yyyy”.  
    
 Only positive value should be given as input to the days to be added,  else print “n value is negative”.  If the date format is not “mm/dd/yyyy” , then print “Invalid date format” .  
    
 Example : 5 days after “10/21/2009” is “10/26/2009”.  
    
 Include a class UserProgramCode with a static method addDays which accepts a String and an Integer. The return type (String) should return the final date as String or it would return "-1" if the day value is negative or it would return "-2" if the date is not as per the given format.  
 Create a Class Program which would be used to accept a String, and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of a String and an integer, where the String corresponds to the input date and the integer corresponds to the number of days.  
 Output consists of a String.  
    
 Refer sample output for formatting specifications.  
    
 Sample Input 1:  
 10/21/2009  
 5  
 Sample Output 1:  
 10/26/2009  
    
 Sample Input 2:  
 10/21/2009  
 -5  
 Sample Output 2:  
 n value is negative  
    
 Sample Input 3:  
 40/21/2009  
 5  
 Sample Output 3:  
 Invalid date format

class Program

{

static void Main(string[] args)

{

string s=Console.ReadLine();

int i=Convert.ToInt16(Console.ReadLine());

string s1 = UserProgramCode.addDays(s,i);

if (s1 == "-1")

Console.WriteLine("n value is negative");

else if (s1 == "-2")

Console.WriteLine("Invalid date format");

Console.WriteLine(s1);

Console.ReadLine();

}

}

class UserProgramCode

{

public static string addDays(string s,int a)

{

string format = "MM/dd/yyyy";

DateTime dt;

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

if (!b)

return "-2";

if (a < 0)

return "-1";

dt=dt.AddDays(a);

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

}

}

**149. Check Anagrams** A anagram is a word or a phrase that can be created by rearranging the letters of another given word or phrase. We ignore white spaces and letter case. All letters of 'Desperation' can be rearranged to the phrase 'A Rope Ends It'. Write a program to check whether the 2 given strings are anagrams or not. Business Rules : 1. If there are any special characters (Space is not considered as special character) in either of the input strings, then store FALSE in the output variable. Create a class named UserProgramCode that has the following static method   
 public static bool checkAnagram(string input1,string input2) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 Input consists of 2 strings.  
 Output is either “TRUE” or “FALSE”.  
 Refer business rules and sample output for formatting specifications. Sample Input 1 : tea eat    
 Sample Output 1 :  
 TRUE Sample Input 2 : Desperation A Rope Ends It    
 Sample Output 2 :  
 TRUE

using System;

class Program

{

public static void Main( string[] args )

{

string str1 = Console.ReadLine();

string str2 = Console.ReadLine();

Console.WriteLine(UserProgramCode.checkAnagram(str1, str2).ToString().ToUpper());

}

}

using System;

class UserProgramCode

{

public static bool checkAnagram(string str1, string str2)

{

// fill your code here

str1 = str1.ToLower();

str1 = str1.Replace(" ", "");

str2 = str2.ToLower();

str2 = str2.Replace(" ", "");

foreach (char item in str1)

{

if (!char.IsLetter(item))

{

return false;

}

}

foreach (char item in str2)

{

if (!char.IsLetter(item))

{

return false;

}

}

foreach (char c in str1)

{

int ix = str2.IndexOf(c);

if (ix == -1)

return false;

}

return true;

}

}

**150. GCD – Array**  
    
 Given an array of integers as input, write a program to find the Greatest Common Divisor for all the integer elements present in the input. Greatest Common Divisor also known as the greatest common factor (gcf), of two or more integers is the largest positive integer that divides the numbers without a remainder. Business Rule: 1. If the input array contains any value less than 1, assign -1 to the output1 variable. Create a class named UserProgramCode that has the following static method   
 public static int greatestCommonDivisor(int[] input1)  
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.  
 The next 'n' lines of input consist of elements in the input array.  
 Output is an integer.  
 Refer business rules and sample output for formatting specifications.  
    
 Sample Input 1 : 4 24 12 20 8    
 Sample Output 1 :  
 4 Sample Input 2 : 4 2 4 8 -6    
 Sample Output 2 :  
 -1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace gcd

{

class UserProgramCode

{

public static int greatestCommonDivisor(int[] input1)

{

int count = 1;

int[] ni = new int[67];

int[] na = new int[67];

int f = 0;

int k = 0;

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

{

if (input1[i] < 0)

{

return -1;

}

}

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

{

for (int j = 1; j <= input1[i]; j++)

{

if (input1[i] % j == 0)

{

ni[k] = j;

k++;

}

}

}

Array.Resize(ref ni, k);

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

{

count = 1;

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

{

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

{

count++;

}

}

if (count == input1.Length)

{

na[f] = ni[i];

f++;

}

}

int ans = na.Max();

return ans;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace gcd

{

class Program

{

static void Main(string[] args)

{

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

int[] nn = new int[n];

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

{

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

}

int op = UserProgramCode.greatestCommonDivisor(nn);

Console.WriteLine(op);

Console.ReadLine();

}

}

}