**1. Time Validation**

**QUESTION:**

Time Validation  
  Write code to validate time using the following rules:  
 Business rules:  
 - It should be a valid time in 12 hrs format  
 - It should have case insensitive AM or PM  
 -The time as input in the following format 'hh:mm am' or 'hh:mm pm'  
 Example:   
 input = 09:59 pm  
 output = Valid time format  
    
 Include a class UserProgramCode with static method validateTime  which accepts the String.The return type should be interger.  
 Create a class Program which would get the input and call the static method validateTime present in the UserProgramCode.  
    
 If the given time is as per the given business rules return 1 else return -1.If the method returns 1 then print "Valid time format" else print "Invalid time format" in Program.  
    
 Input and Output Format:  
 The input time  will be a string  
 Output  will be a string.("Valid time format" or "Invalid time format").  
    
 Sample Input 1:  
 09:59 pm  
    
 Sample Output 1:  
 Valid time format

**ANSWER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace levelI\_02

{

class Program

{

static void Main(string[] args)

{

string str = Console.ReadLine();

int ans = UserProgramCode.validateTime(str);

if (ans == 1)

Console.WriteLine("Valid time format");

else if (ans == -1)

Console.WriteLine("Invalid time format");

}

}

class UserProgramCode

{

public static int validateTime(string str)

{

int hr, min;

hr = int.Parse(str.Substring(0, 2));

min = int.Parse(str.Substring(3, 2));

string suf = str.Substring(5, 3);

if (hr > 12 || min > 60 || suf != " am" && suf != " pm")

return -1;

else

return 1;

}

}

}

**2. Sum Largest Numbers In Range**

**QUESTION:**

Sum Largest Numbers In Range  
    
 Given an array of integer as input1 which falls under the range 1-100, write a program to find the largest numbers from input1 which would fall in the given range 1-10 , 11-20, 21-30, 31-40, …… till 91-100. Now find their sum and print the sum. Business Rules: 1. If the given input array contains any negative number then print -1. 2. If any element is equal to zero or greater than 100 then print -2. 3. In case the array of integer satisfies both business rule 1 as well as 2 then print -3. 4. In case of duplicate numbers eliminate the duplicate number and follow all other steps for calculation of the largest number. Create a class named UserProgramCode that has the following static method   
 public static int largestNumber(int[] input1)  
    
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:    
 The first line of the input consists of an integer n, that corresponds to the size of the array.  
 The next 'n' lines of input consist of integers that correspond to the elements in the array.  
 Output is an integer.  
 Refer business rules and sample output for output format.  
    
 Sample Input 1: 7 13 18 26 34 58 65 54

Sample Output 1 :  
 201 Sample Input 2 :  
 5 -1 19 15 18 101  
 Sample Output 2 :  
 -3

**ANSWER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace SumLargestInRange

{

class Program

{

static void Main(string[] args)

{

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

int[] arr = new int[n];

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

{

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

}

int op=UserProgramCode.sumrange(arr);

Console.WriteLine(op);

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace sum\_Largest\_in\_range

{

class UserProgramCode

{

public static int sumrange(int[] arr)

{

int li = 0, c = 0, c1 = 0, c2 = 0, c3 = 0, c4 = 0, c5 = 0, c6 = 0, c7 = 0, c8 = 0, c9 = 0;

int s = 0;

int count = 0;

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

{

if ((arr[i] < 0))

{

count++;

}

if ((arr[i] == 0) || (arr[i] > 100))

{

count++;

}

if (count == 2)

return -3;

}

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

{

if ((arr[i] == 0) || (arr[i] > 100))

{

return -2;

}

else if (arr[i] < 0)

{

return -1;

}

}

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

{

if (arr[i] > 0 && arr[i] <= 10)

{

if (c == 0)

{

s = s + arr[i];

}

else

{

if (arr[li] < arr[i])

{

s = s - arr[li];

s = s + arr[i];

}

}

c++;

li = i;

}

else if (arr[i] > 10 && arr[i] <= 20)

{

if (c1 == 0)

{

li = 0;

s = s + arr[i];

}

else

{

if (arr[li] < arr[i])

{

s = s - arr[li];

s = s + arr[i];

}

}

c1++;

li = i;

}

else if (arr[i] > 20 && arr[i] <= 30)

{

if (c2 == 0)

{

li = 0;

s = s + arr[i];

}

else

{

if (arr[li] < arr[i])

{

s = s - arr[li];

s = s + arr[i];

}

}

c2++;

li = i;

}

else if (arr[i] > 30 && arr[i] <= 40)

{

if (c3 == 0)

{

li = 0;

s += arr[i];

}

else

{

if (arr[li] < arr[i])

{

s = s - arr[li];

s = s + arr[i];

}

}

c3++;

li = i;

}

else if (arr[i] > 40 && arr[i] <= 50)

{

if (c4 == 0)

{

li = 0;

s = s + arr[i];

}

else

{

if (arr[li] < arr[i])

{

s = s - arr[li];

s = s + arr[i];

}

}

c4++;

li = i;

}

else if (arr[i] > 50 && arr[i] <= 60)

{

if (c5 == 0)

{

li = 0;

s = s + arr[i];

}

else

{

if (arr[li] < arr[i])

{

s = s - arr[li];

s = s + arr[i];

}

}

c5++;

li = i;

}

else if (arr[i] > 60 && arr[i] <= 70)

{

if (c6 == 0)

{

li = 0;

s = s + arr[i];

}

else

{

if (arr[li] < arr[i])

{

s = s - arr[li];

s = s + arr[i];

}

}

c6++;

li = i;

}

else if (arr[i] > 70 && arr[i] <= 80)

{

if (c7 == 0)

{

li = 0;

s = s + arr[i];

}

else

{

if (arr[li] < arr[i])

{

s = s - arr[li];

s = s + arr[i];

}

}

c7++;

li = i;

}

else if (arr[i] > 80 && arr[i] <= 90)

{

if (c8 == 0)

{

li = 0;

s = s + arr[i];

}

else

{

if (arr[li] < arr[i])

{

s = s - arr[li];

s = s + arr[i];

}

}

c8++;

li = i;

}

else if (arr[i] > 90 && arr[i] <= 100)

{

if (c9 == 0)

{

li = 0;

s = s + arr[i];

}

else

{

if (arr[li] < arr[i])

{

s = s - arr[li];

s = s + arr[i];

}

}

c9++;

li = i;

}

}

return s;

}

}

}

**3. Next Consonant or Vowel**

**QUESTION:**

Next Consonant or Vowel  
    
 Given an input String, write a program to replace all the vowels of the given string with the next consonant and replace all consonants with the next available vowel. Business Rule: 1. If the input string contains any number or any special characters, print 'Invalid input'. 2. The input is case sensitive. Please ensure that each character in the output has exactly the same case as the input string. Create a class named UserProgramCode that has the following static method   
 public static string nextString(String input1)  
    
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:  
 Input consists of a string.  
 Output consists of a string. Refer business rules and sample output for the format.  
    
 Sample Input 1 : zebRa  
    
 Sample Output 1 : afeUb Sample Input 2 : cat@rat/123  
    
 Sample Output 2 : Invalid input

**ANSWER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Reflection;

namespace ConsoleApplication2

{

class Program

{

static void Main(string[] args)

{

string str = Console.ReadLine();

Console.WriteLine(UserProgramCode.nextString(str));

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Reflection;

using System.Collections;

namespace nextConsonentorVowel

{

class UserProgramCode

{

public static string nextString(string str)

{

ArrayList vowel\_small = new ArrayList();

ArrayList vowel\_caps = new ArrayList();

vowel\_small.Add('a');

vowel\_small.Add('e');

vowel\_small.Add('i');

vowel\_small.Add('o');

vowel\_small.Add('u');

vowel\_caps.Add('A');

vowel\_caps.Add('E');

vowel\_caps.Add('I');

vowel\_caps.Add('O');

vowel\_caps.Add('U');

char[] inp = str.ToCharArray();

char[] out1 = new char[str.Length];

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

{

if (!Char.IsLetterOrDigit(str[i]))

return "invalidinput";

}

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

{

if (vowel\_caps.Contains(inp[i]) || vowel\_small.Contains(inp[i]))

{

char ch = (char)((int)inp[i] + 1);

out1[i] = ch;

}

else if (inp[i] == 90 || inp[i] == 122)

{

if (inp[i] == 90)

out1[i] = 'A';

else

out1[i] = 'a';

}

else

{

if (inp[i] >= 65 && inp[i] <= 90)

{

if (inp[i] > 85)

out1[i] = 'A';

else

{

foreach (char che in vowel\_caps)

{

if (che > inp[i])

{

out1[i] = che;

break;

}

}

}

}

else

{

if (inp[i] > 117)

out1[i] = 'a';

else

{

foreach (char che in vowel\_small)

{

if (che > inp[i])

{

out1[i] = che;

break;

}

}

}

}

}

}

string output = null;

foreach (char c in out1)

{

output += c.ToString();

}

return output;

}

}

}

**4. Donations**

**QUESTION:**

Donations  
    
 Given 2 inputs,string array input1 and integer input2.The usercodes,locations and donations are appended as one element and stored in input1 in the following format, ABCDEFGHI- here the ABC represents the usercode ,DEF represents the location and GHI represents the donation amount. Write a program to find the total amount donated by the users who have the same location code given in input2 integer value. Business rule: 1) If the string array contains any duplicates, then print -1. 2) If the string array contains any special characters, then print -2. Create a class named UserProgramCode that has the following static method   
 public static int getDonation(string[] input1, int input2)  
    
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.  
    
 Input and Output Format:    
 The first line of the input consists of an integer, n that corresponds to the number of elements in the string array.  
 The next 'n' lines of input consists of strings that correspond to elements in the string array.  
 The next line of the input consists of an integer that corresponds to the location code.  
 Refer business rules and sample output for output format.  
    
    
 Sample Input 1 :  
 4 123111241 124222456 145111505 124553567 111    
 Sample Output 1 : 746 Sample Input 2 : 4 123111241 124222456 124222456 124553567 111    
 Sample Output 2 : -1

**ANSWER:**

using System;

using System.Text.RegularExpressions;

namespace code1

{

class Program

{

static void Main(String[] args)

{

int n;

Regex reg = new Regex(@"([A-Za-z0-9])$");

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

String[] input1 = new String[n];

int input2;

int output;

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

{

input1[i] = Console.ReadLine();

if (!reg.IsMatch(input1[i]))

{

Console.WriteLine("-2"); return;

}

}

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

{

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

{

if(input1[i].Equals(input1[j]))

{ Console.WriteLine("-1");}

}

}

input2 = int.Parse(Console.ReadLine());

output = UserMainCode.getDonation(input1, input2);

Console.WriteLine(output);

}

}

}

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

using System;

public class UserMainCode

{

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

{

String temp;

int n=input1.Length;

int output=0,don;;

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

{

temp = input1[i].Substring(3, 3);

if(int.Parse(temp)==input2){

don=int.Parse(input1[i].Substring(6,3));

output += don;

}

}

return output;

**}**

**}**

**5. Max Diff in Array**

**QUESTION:**

Max Diff in Array  
 Given an integer input array input,find out the maximum difference between any two elements such that larger element appears after the smaller number in the input array. Business Rules : 1. If any of the given inputs contain any negative number, then print -1. 2. If there is only one element or more than 10 elements in the input array, then print -2. 3. If there are any duplicates in the input array, then print -3.    
 Create a class named UserProgramCode that has the following static method   
 public static int diffIntArray(int[] input1)  
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.  
 The next 'n' lines of input consist of elements in the input array.  
 Output is an integer.  
 Refer business rules and sample output for formatting specifications.  
    
 Sample Input 1 : 7 2 3 10 6 4 8 1    
 Sample Output 1 :  
 8  
    
 [Hint : (Diff between 2 and 10. 10 is larger than 2)] Sample Input 2 : 5 4 5 -20 9 10    
 Sample Output 2 :  
 -1

**ANSWER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Max\_diference

{

class UserProgramCode

{

public static int diffIntArray(int[] input1)

{

int diff=0;

int len =input1.Length;

int max = 0;

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

{

if (input1[i] < 0)

{

return -1;

}

}

if (len < 2 || len > 10)

return -2;

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

{

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

{

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

return -3;

}

}

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

{

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

{

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

{

int m = input1[j] - input1[i];

if (m > max)

max = m;

}

}

}

return max;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Max\_diference

{

class Program

{

static void Main(string[] args)

{

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

int[] arr = new int[n];

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

{

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

}

int op = UserProgramCode.diffIntArray(arr);

Console.WriteLine(op);

Console.ReadLine();

}

}

}

**6. List the Elements A**

**QUESTION:**

List the Elements - A  
 Write a program that accepts integer list and an integer. List all the elements in the list that are smaller than the value of given integer. Print the result in descending order.  
    
 Example:  
 input1: [1,4,7,3,9,15,24]  
 input2: 17  
    
 Output1:[15,9,7,4,3,1]  
    
 Include a class UserProgramCode with static method GetElements() which accepts an integer list and the integer (input2) as input and returns an integer list.  If there is no element found in input1, then store -1 to the first element of output list. Create a class Program which would get the input and call the static method GetElements() present in the UserProgramCode. If there is no such element in the input list, print "No element found".  
    
 Input and Output Format:  
 Input consists of n+2 integers. The first integer corresponds to n, the number of elements in the array. The next'n' integers correspond to the elements in the array.  
    
 The last input is an integer.  
    
 Output is an integer list or the string "No element found".  
    
 Sample Input 1: 7  
 1 4  
 7  
 3  
 9  
 15  
 24  
 17  
 Sample Output 1: 15 9 7 4 3 1 Sample Input 2: 6 5 9 3 4 16 21 9 Sample Output 2: 5 4 3  

**ANSWER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace List\_the\_elements

{

class Program

{

static void Main(string[] args)

{

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

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

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

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

{

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

}

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

output= UserProgramCode.GetElements(a,n,chk);

if (output[0] == -1)

{

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

}

if (output[0] != -1)

{

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

{

Console.WriteLine(output[i]);

}

}

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace List\_the\_elements

{

class UserProgramCode

{

public static List<int> GetElements(List<int> a,int n,int chk)

{

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

int count =0;

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

{

if (a[i] < chk)

{

temp.Add(a[i]);

count = 1;

}

}

temp.Sort();

temp.Reverse();

if (count == 0)

{

temp.Add(-1);

}

return temp;

}

}

}

**7. Is – Is Not**

**QUESTION:**

Is – Is Not  
    
 Write a program to read a String and to replace every appearance of the word "is" by "is not". If the word "is" is immediately preceeded or followed by a letter no change should be made to the string. Print the final string.  
    
 Example:   
 input = This is just a misconception  
 output = This is not just a misconception  
    
 Include a class UserProgramCode with a static method negativeString which accepts a string. The return type (String) should return the final output.  
 Create a Class Program which would be used to accept a string input, and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of a string.  
 Output consists of a string.  
 Refer sample output for formatting specifications.  
    
 Sample input 1:  
 This is just a misconception  
 Sample Output 1:  
 This is not just a misconception

**ANSWER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Reflection;

namespace ConsoleApplication2

{

class Program

{

static void Main(string[] args)

{

string s = Console.ReadLine();

string output = UserProgramCode.negativeString(s);

Console.WriteLine(output);

}

}

}

class UserProgramCode

{

public static string negativeString(string str)

{

string neg\_string = null;

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

StringBuilder sb = new StringBuilder();

for (int i=0;i<str1.Length;i++)

{

if (str1[i].Equals("is"))

{

sb.Append("is not ");

}

else

{

sb.Append(str1[i]+" ");

}

}

neg\_string = sb.ToString();

return neg\_string;

}

}

**8. convertRomanToDecimal**

**QUESTION:**

Convert Roman to Decimal   
    
 Write a program to convert the given roman number to decimal number.  
    
 Example : Input string: XVII Output variable:10+5+1+1 = 17  
 The input string should contain only the alphabets given below (in upper case) . Valid alphabets are I, V, X, L, C, D, and M 'I': The corresponding value = 1 'V': The corresponding value = 5 'X': The corresponding value =10 'L': The corresponding value = 50 'C': The corresponding value = 100 'D': The corresponding value = 500 'M': The corresponding value =1000  
    
 Include a class UserProgramCode with a static method convertRomanToDecimal that accepts a string and returns an integer. The method returns -1 if the input string is not valid.  
    
 Create a Class Program which would be used to read the string and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of a string.  
 Output consists of an integer.  
 Sample Input 1: XII Sample Output 1: 12 Sample Input 2: DCL Sample Output 2: 650  
    
 Sample Input 3: DCA Sample Output 3: -1

**ANSWER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Roman\_to\_decimal

{

class UserProgramCode

{

public static int RomanToDecimal(string str)

{

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

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

int sum = 0;

romanList.Add("I");

romanList.Add("V");

romanList.Add("X");

romanList.Add("L");

romanList.Add("C");

romanList.Add("D");

romanList.Add("M");

intList.Add(1);

intList.Add(5);

intList.Add(10);

intList.Add(50);

intList.Add(100);

intList.Add(500);

intList.Add(1000);

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

{

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

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

else

return -1;

}

return sum;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Roman\_to\_decimal

{

class Program

{

static void Main(string[] args)

{

string str = Console.ReadLine();

int op = UserProgramCode.RomanToDecimal(str);

Console.WriteLine(op);

Console.ReadLine();

}

}

}

**9. Count of Elements**

**QUESTION:**

113013  
 Count of Elements  
 Write a program that gets the count of elements in input1 list that starts with the character passed in input2 irrespective of case. Print the count.  
    
 Example:   
 input1: ['abc','Apple','Mango']  
 input2: a  
    
 Output1:  
 2  
    
 Business Rule:  
 1. If there is no element that start with the given char in input1, then return -1.  
 2. Only alphabets should be given in input1 string else return -2.  
    
 Include a class UserProgramCode with a static method GetCount which accepts the size of the string array, string array and a character. The return type (Integer) should return count. Follow the Business rules.  
 Create a Class Program which would be used to accept the size of the array, the array elements and a character, and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of an integer, which corresponds to the size of the array, a string list, and a character.  
 Output consists of an Integer(final count), or a String(“No elements Found” if -1 is returned or “Only alphabets should be given” if -2 is returned.  
 Refer sample output for formatting specifications.  
    
 Sample Input 1:  
 3  
 abc  
 Apple  
 Mango  
 a  
 Sample Output 1:  
 2  
    
 Sample Input 2:  
 2  
 goods  
 bads  
 a  
 Sample Output 2:  
 No elements Found  
    
 Sample Input 3:  
 2  
 good$  
 bad$  
 a  
 Sample Output 3:  
 Only alphabets should be given

**ANSWER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Reflection;

namespace ConsoleApplication2

{

class Program

{

static void Main(string[] args)

{

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

string[] str = new string[n];

if (n > 0)

{

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

{

str[i] = Console.ReadLine();

}

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

int output = UserProgramCode.getCount(n, str, c);

if (output > 0)

{

Console.WriteLine(output);

}

else if (output == -1)

{

Console.WriteLine("No elements Found");

}

else if (output == -2)

{

Console.WriteLine("Only alphabets should be given");

}

}

}

}

}

class UserProgramCode

{

public static int getCount(int size,string[] str,char c)

{

int count=0;

char c\_cap = c;

char c\_small = c;

Regex reg = new Regex(@"^([A-Za-z]{1,})$");

foreach (string s in str)

{

if (!reg.IsMatch(s))

return -2;

string ch = c.ToString();

if (c >= 97 && c <= 122)

{

c\_cap = (char)((int)(c) - 32);

}

else if (c >= 65 && c <= 90)

{

c\_small = (char)((int)(c) + 32);

}

char[] inp = s.ToCharArray();

if (c\_small == inp[0]||c\_cap==inp[0])

count++;

}

if (count >= 1)

{

return count;

}

else

return -1;

}

}

**10. Digit Sum in String Array**

**QUESTION:**

Digit Sum in String Array  
 Given a String array as input. Each element in this array may contain alphabets or digits. Develop code to add all the digits in every string and print the sum as an int. If two digits appear simultaneously do not consider it as one number. Ex- For 'Hyderabad 21' consider 2 and 1 as two digits instead of 21 as a number. Business Rules : 1. If there are any special characters in the input strings, then print -1. Example 1: input1: 5 input2: AAA1 B2B 4CCC A5 ABCDE output1:1+2+4+5=12 Example 2: input1 : 3 input2 : 12 C23 5CR2 output1 : 1+2+2+3+5+2 = 15    
 Create a class named UserProgramCode that has the following static method   
 public static int sumOfDigits(string[] input1)  
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 The first line of the input consists of an integer, n that corresponds to the number of elements in the input string array.  
 The next 'n' lines of input consist of elements in the input string array.  
 Refer sample output for formatting specifications.    
 Sample Input : 5 AAA1 B2B 4CCC A5 ABCDE  
    
 Sample Output :  
 12  
  

**ANSWER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace SumOfdigitinString

{

class Program

{

static void Main(string[] args)

{

string s = Console.ReadLine();

int result = UserProgramCode.sumofdigitsinstring(s);

Console.WriteLine(result);

Console.ReadLine();

}

public static int sumofdigitsinstring(string input1)

{

int temp, sum = 0;

StringBuilder sb = new StringBuilder();

// string input1 = "app123e";

char[] ch = input1.ToCharArray();

foreach (char c in ch)

{

if (char.IsDigit(c))

{

sb.Append(c);

}

}

string s = sb.ToString();

int n = int.Parse(s);

while (n > 0)

{

temp = n % 10;

sum = sum + temp;

n = n / 10;

}

return sum;

}

}

}

**11. Calculate the Efficiency**

**QUESTION:**Calculate the Efficiency  
 In a company, worker efficiency is determined on the basis of the time required for a worker to complete a particular job.If the time taken by the worker is input, then display the efficiency of the worker.  
    
 If time taken is 1 to 3 hours then the worker is said to be "Highly efficient".  
 If time taken is more than 3 hrs and less than or equal to 4 hours then efficiency level is "Improve speed"  
 If time taken is more than 4 hours and less than or equal to 5 hours then efficiency level is "Need Training" to improve speed.  
 If the time taken is more than 5 hours, then the worker has to "Leave the company".   
 otherwise it should return Invalid Input  
 Include a class UserProgramCode with static method EfficiencyChecker which accepts float.The return type is String.  
    
 Create a class Program which would get the input and call the static method EfficiencyChecker present in the UserProgramCode.  
    
 Input output format The input consists a float. The output consists the String. Sample Input 1: 5.0 Sample Output 1: Need Training Sample Input 2: 10.5 Sample Output 2: Leave the company Sample Input 2: -2 Sample Output 2: Invalid Input

**ANSWER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace progm51

{

class Program

{

static void Main(string[] args)

{

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

string outp=UserProgramCode.EfficiencyChecker(inp);

Console.WriteLine(outp);

}

}

class UserProgramCode

{

public static string EfficiencyChecker(decimal a)

{

if ((a >= 1) && (a <= 3))

{

return "Highly efficient";

}

else if ((a > 3) && (a <= 4))

{

return "Improve speed";

}

else if ((a > 4) && (a <= 5))

{

return "Need Training";

}

else if (a > 5)

{

return "Leave the company";

}

else

{

return "Invalid Input";

}

}

}

}

**12. Sum of Cube**

**QUESTION:**

Calculate the sum of cube  
    
    
    
 Write a program to find the sum of the cube of first 'n' natural numbers.    
    
 Example:  
 input = 5  
 output = 225  
    
 Include a class UserProgramCode with a static method sumOfCube() that accepts an integer and returns an integer . If the input is not a natural number, return -1. Create a class Program which would get the input and call the static method sumOfCube() present in the UserProgramCode.  
    
 Input and Output Format:  
    
 Input is an integer that corresponds to n  
 Output is an integer (Sum of cubes) or if the given input n is not a natural number then print “The input is not a natural number”  
    
 Sample Input 1:  
 5  
 Sample Output 1:  
 225  
    
 Sample Input 2:  
 -1  
 Sample Output 2:  
 The input is not a natural number

**ANSWER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication2

{

class Program

{

static void Main(string[] args)

{

{

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

int sl = UserProgramCode.sumOfCube(n);

if (sl == -1)

Console.WriteLine("The input is not a natural number");

else

Console.WriteLine(sl);

Console.ReadLine();

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication2

{

class UserProgramCode

{

public static int sumOfCube(int n)

{

if (n < 0)

return -1;

int sum = 0;

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

sum = sum + (i \* i \* i);

return sum;

}

}

}

**13. Vehicle Mileage**

**QUESTION:**

Vehicle Mileage   
 Write a program to find the mileage of the vehicle given the bike caliber (input1) and the number of years (input2) the vehicle is being used as inputs and print the output as given in the sample output. The ideal mileage details given by the manufacturer are as follows : if CC is between 100 to 125 mileage is 75 if CC is between 126 to 135 mileage is 70 if CC is between 136 to 150 mileage is 60 if CC is between 151 to 200 mileage is 50 if CC is between 201 to 220 mileage is 35 But with the years of usage of the vehicle, the promised mileage is reduced as follows : 1.Vehicle Usage till 2 years, the mileage is reduced by 10% than the promised. 2.Vehicle Usage more than 2 years till 4 years, the mileage is reduced by 15% than the promised. 3.Vehicle Usage more than 4 years, the mileage is reduced by 20% than the promised. The output to be printed is  
 The mileage of the bike is xyz  
 where xyz is the calculated mileage. Business Rules :- 1) If the bike caliber given is negative number or below 100, then print 'Invalid caliber'. 2) If the number of years given is a negative number or more than 20 years, then print 'Invalid years'. Create a class named UserProgramCode that has the following static method   
 public static double findMileage(int input1, int input2)  
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 Input consists of 2 integers.  
 The first line of the input corresponds to bike calibre and the second line of the input corresponds to   
 number of years.  
 Refer sample output and business rule for output formatting specifications.  
    
 Sample Input 1 : 100 1  
 Sample Output 1 :  
 The mileage of the bike is 67.5 Sample Input 2 : 90 1  
 Sample Output 2 :  
 Invalid caliber  
  

**ANSWER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Vechile

{

class Program

{

static void Main(string[] args)

{

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

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

double op = UserProgramCode.findmileage(cc, year);

if (op == -1)

{

Console.WriteLine("Invalid Caliber");

}

else

Console.WriteLine("The mileage of the bike is {0}",op);

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Vechile

{

class UserProgramCode

{

public static double findmileage(int cc, int year)

{

double mileage = 0;

if (cc >= 100 && cc <= 125)

{

if (year < 2)

{

mileage = 75 - (75 \* 0.1);

}

if (year > 2 && year <= 4)

{

mileage = 75 - (75 \* 0.15);

}

if (year > 4)

{

mileage = 75 - (75 \* 0.20);

}

}

else if (cc >= 126 && cc <= 135)

{

if (year < 2)

{

mileage = 70 - (70 \* 0.1);

}

if (year > 2 && year <= 4)

{

mileage = 70 - (70 \* 0.15);

}

if (year > 4)

{

mileage = 70 - (70 \* 0.20);

}

}

else if (cc >= 136 && cc <= 150)

{

if (year < 2)

{

mileage = 60 - (60 \* 0.1);

}

if (year > 2 && year <= 4)

{

mileage = 60 - (60 \* 0.15);

}

if (year > 4)

{

mileage = 60 - (60 \* 0.20);

}

}

else if (cc >= 151 && cc <= 200)

{

if (year < 2)

{

mileage = 50 - (50 \* 0.1);

}

if (year > 2 && year <= 4)

{

mileage = 50 - (50 \* 0.15);

}

if (year > 4)

{

mileage = 50 - (50 \* 0.20);

}

}

else if (cc >= 201 && cc <= 220)

{

if (year < 2)

{

mileage = 35 - (35 \* 0.1);

}

if (year > 2 && year <= 4)

{

mileage = 35 - (35 \* 0.15);

}

if (year > 4)

{

mileage = 35 - (35 \* 0.20);

}

}

else

return -1;

return mileage;

}

}

**14. Get Common Elements**

**QUESTION:**

Get Common Elements  
 Write a program that accepts two lists and finds out the elements that are common in both of the given input lists and consolidates the common elements in another list. Business Rule: Only positive number should be given in input List. Else print -1 in the output. Create a class named UserProgramCode that has the following static method  
 public static List<int> getCommonItems(List<int> input1, List<int> input2)  
 Create a class named Program that accepts the 2 input lists and calls the static method present in the UserProgramCode.  
 Input and Output Format:  
 The first line of the input consists of an integer that corresponds to m, the number of elements in the first list. The next 'm' lines consist of elements in the first list.  
 The next line of the input consists of an integer that corresponds to n, the number of elements in the second list. The next 'n' lines consist of elements in the second list.  
 Output consists of the list that contains the common elements in both the lists. If any of input list elements are negative, print -1.  
 Sample Input : 5 5 6 7 8 1 4 5 2 3 1  
 Sample Output : 5 1

**ANSWER:**

-----Get Common Elements----

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace getcommonelements

{

class Program

{

static void Main(string[] args)

{

int m, n,i,k,c;

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

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

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

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

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

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

{

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

listm.Add(k);

}

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

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

{

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

listn.Add(c);

}

listo = UserProgramCode.getCommonItems(listm, listn);

final = listo.Distinct().ToList();

foreach (int t in final)

{

Console.WriteLine(t);

}

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace getcommonelements

{

class UserProgramCode

{

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

{

int len1, len2;

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

len1 = input1.Count;

len2 = input2.Count;

foreach (int i in input1)

{

foreach (int k in input2)

{

if (i == k)

{

p.Add(i);

}

}

}

return p;

}

}

}

**15. Find largest digit in a given number**

**QUESTION:**

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

**ANSWER:**

19)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Fwd\_Prgs

{

public class UserProgramCode

{

public static int findLargestDigit(int num)

{

if (num > 0)

{

int temp, res = 0;

while (num > 0)

{

temp = num % 10;

if (temp > res)

res = temp;

num = num / 10;

}

return res;

}

else

return -1;

}

}

class Program

{

static void Main(string[] args)

{

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

int res = UserProgramCode.findLargestDigit(n);

if(res!=-1)

Console.WriteLine(res);

else

Console.WriteLine("The number should be a positive number");

}

}

}

**16. Employee Designation**

**QUESTION:**

Employee Designation  
 Given an input1 string array in the format {Employee1, Designation, Employee2, Designation, Employee3, Designation, and so on... } and a string input2, write a program to fetch the employee names from input1 based on input2 (designation) value and assign it in an output array and print the array. Case sensitivity can be ignored. Business rule: 1) If input1 or input2 contains any special characters, then print 'Invalid Input' 2) If input1 does not contain the designation in input2, then print 'No employee for ABC designation' where ABC is the Input2 value. 3) If all the employees belong to the same designation, then print 'All employees belong to same ABC designation' where ABC is the Input2 value. Example 1: input1: Ram Manager Ganesh Developer Srijith Developer input2: Developer output : Ganesh Srijith Example 2: Input 1: Manish BiDeveloper Babu Manager Rohit Associate Input 2: System Analyst Output1: No employee for System Analyst designation Create a class named UserProgramCode that has the following static method   
 public static string[] getEmployee(string[] input1, string input2)  
    
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.  
    
 Input and Output Format:    
 The first line of the input consists of an integer, n that corresponds to the number of elements in the string array.  
 The next 'n' lines of input consists of strings that correspond to elements in the string array.  
 The next line of the input consists of a string that corresponds to the Designation.  
 Refer business rules and sample output for output format.  
    
    
 Sample Input 1:  
 6 Ram Manager Ganesh Developer Srijith Developer Developer  
 Sample Output 1:  
 Ganesh Srijith  
    
 Sample Input 2:  
 6 Manish BiDeveloper Babu Manager Rohit Associate System Analyst    
 Sample Output 2: No employee for System Analyst designation  

**ANSWER:**

Employee designation

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace EmployeeDesgination

{

class Program

{

static void Main(string[] args)

{

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

string[] str = new string[n];

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

{

str[i] = Console.ReadLine();

}

string find = Console.ReadLine();

string[] op = UserProgramCode.getEmployee(str, find);

if (op.Length == str.Length / 2)

{

Console.WriteLine();

}

if (op.Length == 0)

{

Console.WriteLine();

}

foreach (string item in op)

{

Console.WriteLine(item);

}

Console.ReadLine();

}

}

}

userprogram

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace EmployeeDesgination

{

class UserProgramCode

{

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

{

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

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

{

if (input2.ToLower() == input1[i].ToLower())

{

list.Add(input1[i-1]);

}

}

return list.ToArray();

}

}

}

**17.** **Replace String**

**QUESTION:**

Replace String  
    
 Write a program to form a new string by replacingeach character in the n'th word of the input string with given special character. Display resultant string in lowercase.  
    
 Example:  
    
 Input1: Hi are you fine Ram  
 Input2: 5  
 Input3: \*  
 Output:hi are you fine \*\*\*  
    
 Business Rules:  
    
 1. Only alphabets should be given in input1 string Else return "-1" from the method and print “Invalid String” in Main.  
 2. Only positive number should be given for input2 Else return "-2" from the method and print “Number not positive” in Main.  
 3. Only special characters should be given for input3 Else return "-3" from the method and print “Character not a special character” in Main.   
    
 Include a class UserProgramCode with a static method replaceString which accept a String, an integer and a character. The return type (String) should return the Final String.  
 Create a Class Program which would be used to accept a String, an integer and a character , and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of a String, an integer and a character, where String corresponds to the input string, the integer corresponds to the word number and the character values corresponds to the change character.  
 Output consists of a String.  
    
 Refer sample output for formatting specifications.  
    
 Sample Input 1:  
 Hi are you fine Ram  
 5  
 \*  
 Sample Output 1:  
 hi are you fine \*\*\*  
    
 Sample Input 2:  
 Hi @re you fine Ram  
 5  
 \*  
 Sample Output 2:  
 Invalid String  
    
 Sample Input 3:  
 Hi are you fine Ram  
 -5  
 \*  
 Sample Output 3:  
 Number not positive  
    
 Sample Input 4:  
 Hi are you fine Ram  
 5  
 o  
 Sample Output 4:  
 Character not a special character

**ANSWER:**

REPLACE STRING

using System;

class Program

{

public static void Main( string[] args )

{

string inputWord=Console.ReadLine();

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

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

string result=UserProgramCode.replaceString(inputWord,position,ch);

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

Console.WriteLine("Invalid String");

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

Console.WriteLine("Number not positive");

else if(result.Equals("-3"))

Console.WriteLine("Character not a special character");

else

Console.WriteLine(result);

}

}

using System;

class UserProgramCode

{

public static string replaceString(string inputWord, int position, char ch)

{

string inputWord1=inputWord.ToLower();

foreach (Char z in inputWord)

{

if (!(Char.IsLetterOrDigit(z) || Char.IsWhiteSpace(z)))

{

return "-1";

}

}

if (position <= 0)

{

return "-2";

}

if ((Char.IsLetterOrDigit(ch)) || Char.IsWhiteSpace(ch))

{

return "-3";

}

else

{

string[] A = inputWord1.Split(' ');

string b = string.Copy(A[position - 1]);

char[] B = b.ToCharArray();

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

{

B[i] = ch;

}

string c = new string(B);

A[position - 1] = c;

string d = string.Join(" ", A);

return d;

}

}

}

**18. Form String**

**QUESTION:**

Form String  
 Given a String array and an int 'n', write a program to perform the following operations: 1) Pick nth character from each String element in the String array and form a new String. 2) If nth character not available in a particular String in the array consider $ as the character. 3) Print the new String. Business Rules : 1. If there are any special characters in the input strings, then print -1.    
 Create a class named UserProgramCode that has the following static method  
 public static string formString(string[] input1,int input2)  
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 The first line of the input consists of an integer 'k' that corresponds to the number of elements in the string array.  
 The next 'k' lines of the input consists of strings that correspond to the elements in the string array.  
 The next line of the input consists of an integer that corresponds to n.  
 Refer sample output for formatting specifications.  
 Sample Input : 4 ABC XYZ EFG MN 3 Sample Output : CZG$  

**ANSWER:**

1.Form string

FIRST TRY THIS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace formstr78

{

class Program

{

static void Main(string[] args)

{

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

string[] s = new string[n];

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

{

s[i] =Console.ReadLine();

}

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

string z= form.fun(s, p);

if (z == "-2")

Console.WriteLine("-2");

else

Console.WriteLine(z);

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

namespace formstr78

{

class form

{

public static string fun(string[] a, int f)

{

//StringBuilder sb = new StringBuilder();

StringBuilder sb1 = new StringBuilder();

string p = @"[a-zA-Z]+$";

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

{

if (!Regex.IsMatch(a[i], p))

return "-2";

int len = a[i].Length;

if (len < f)

sb1.Append("$");

else

sb1.Append(a[i].Substring(len-1));

}

string z = sb1.ToString();

return z;

}

}

}

2.

//program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Collections;

namespace trial

{

class Program

{

static void Main(string[] args)

{

String str = Console.ReadLine();

String res = UserProgramCode.getSpecialChar(str);

Console.WriteLine(res);

}

}

}

//UserProgramCode.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace trial

{

class UserProgramCode

{

public static string getSpecialChar(string input1)

{

int num = 0;

StringBuilder sb=new StringBuilder();

int sp = 0;

int len = input1.Length;

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

{

char c = input1[i];

if (char.IsDigit(c))

{

num++;

sb.Append(c);

}

if (c == '#' || c == '$' || c == '%' || c == '&')

{

sp++;

sb.Append(c);

}

//else if (!char.IsLetter(c))

//{

// sb.Append(c);

//}

}

if (num == 0 || sp == 0)

{

return "-1";

}

else

return sb.ToString();

}

}

}

**19. Calculate Commission**

**QUESTION:**

Calculate Commission   Write a program to calculate the commission on given sales as per the following policy. Include a class UserProgramCode with a static method calculateCommission which accepts a float as input.  
 Create a class Program  which would get the input and call the static method calculateCommission present in the UserProgramCode. If the method returns -1, then print 'Invalid Input'.   If sales is less than Rs. 10000/- no commission. If sales is between Rs. 10000/- to Rs. 25000/- commission is 10% of sales. If sales is more than Rs. 25000/- then commission is Rs. 500/- plus 8% of sales amount.   Business Rule : 1. If input is negative number then the method calculateCommission returns -1. 2. Otherwise return a calculated commission. Input and Output format : Input consists of float. Refer sample output for formatting specifications.   Sample Input 1 : 11000 Sample Output 1: 1100 Sample Input 2 : -1000 Sample Output 2 : Invalid Input

**ANSWER:**

70) using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication26

{

class Program

{

static void Main(string[] args)

{

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

int a = UserProgramCode.calculateCommission(s);

if (a == -1)

Console.WriteLine("Invalid Input");

else

Console.WriteLine(a);

Console.ReadLine();

}

}

class UserProgramCode

{

public static int calculateCommission(float a)

{

double com;

double r = Convert.ToDouble(a);

if (r < 0)

{

return -1;

}

else if (r < 10000)

{

return 0;

}

else if (r >= 10000 && r <= 25000)

{

com = r \* 1.0 / 10.0;

return Convert.ToInt32(com);

}

else if (r > 25000)

{

com = 500 + (r \* 8.0/ 100.0);

return Convert.ToInt32(com);

}

return -1;

}

}

}

**20. Count Subsets**

**QUESTION:**

Count Subsets  
    
 Given a method with an integer list as input, Write code to find the number of subsets formed from the given input. Consider any three elements for the input list which forms as one subset. Sum of first two elements must be equal to third element. The number of subsets which satisfy these conditions would be the output that is printed Note: The elements in a subset should satisfy below conditions: 1)Any subset should have only 3 elements 2)The elements in each subset must be distinct Business rule: 1) Print -1 when no such subsets are formed 2) Print -2 if input list consists of negative elements 3) Print -3 when same integer element is repeated twice in the input list. Example: input 5 Input1 1 2 3 4 6 The subsets formed are (1,2,3), (1,3,4), (2,4,6) output = 3 Create a class named UserProgramCode that has the following static method   
 public static int countSubsets(int[] input1)  
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 The first line of the input consists of an integer, n that corresponds to the number of elements in the input array.  
 The next 'n' lines of input consist of elements in the input array.  
 Output consists of an integer.  
 Refer business rules and sample output for formatting specifications.    
 Sample Input 1 : 5 1 2 3 4 6  
    
 Sample Output 2 :  
 3

**ANSWER:**

52.CountSubsets

using System;

class Program

{

public static void Main( string[] args )

{

int arrSize,output;

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

int[] input = new int[arrSize];

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

{

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

}

output = UserProgramCode.countSubsets(input);

Console.WriteLine(output);

Console.Read();

}

}

using System;

class UserProgramCode {

public static int countSubsets(int[] input1)

{

int count = 0,flag=0;

foreach (int l in input1)

{

if (l < 0)

{

flag++;

return -2;

}

}

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

{for(int b=a+1;b<input1.Length;b++)

if (input1[a] == input1[b])

{

flag++;

return -3;

}

}

if (flag == 0)

{

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

{

for (int j = i + 1; j < input1.Length; j++)

{

for (int k = j + 1; k < input1.Length; k++)

{

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

count++;

}

}

}

}

if (count == 0)

return -1;

else

return count;

}

}

**21. Duplicate Characters**

**QUESTION:**

Duplicate Characters  
    
 Write a Program which removes duplicate characters from the string. Your program should read a sentence (string) as input from user and return a string removing duplicate characters. Retain the first occurance of the duplicate character. Assume the characters are case – sensitive.  
 Include a class UserProgramCode with a static method removeDuplicates which accepts a string and returns a string.  
 Create a Class Program which would be used to accept the input string and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of a string with maximum size of 100 characters.   
 Output consists of a single string.  
 Refer sample output for formatting specifications.  
 Sample Input 1:  
 hi this is sample test  
 Sample Output 1:  
 hi tsample    
 Sample Input 2:  
 ABC DEF  
 Sample Output 2:  
 ABC DEF

**ANSWER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace duplicatecharacters

{

class Program

{

static void Main(string[] args)

{

string input;

input = Console.ReadLine();

Class1 c=new Class1();

string output=c.removeDuplicates(input);

foreach (char ch in output)

{

if (ch != '\*')

{

Console.Write(ch);

}

}

//Console.WriteLine(s);

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace duplicatecharacters

{

class Class1

{

public string removeDuplicates(string input)

{

//string[] s=new string[10];

//s=input.Split(

int i,j;

char[] c = new char[100];

c = input.ToCharArray();

int length= c.Length;

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

{

for (j = i + 1; j < length; j++)

{

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

{

c[j] = '\*';

}

}

}

string output = new string(c);

return output;

}

}

}

**22. Digit Sum**

**QUESTION:**

Digit Sum  
 Given a non-negative int n, return the sum of its digits. If sum is greater than 9 repeat the process and calculate the sum once again until the final sum comes to single digit. Example 1: Input=9999 Output: 9 (9+9+9+9 = 36 and 3+6 = 9) Example 2: Input=698 Output: 5 (6+9+8 = 23 and 2+3 = 5)    
 Create a class named UserProgramCode that has the following static method  
 public static int getDigitSum(int input1)  
 Create a class named Program that accepts the input and calls the static method present in the UserProgramCode.    
 Input and Output Format: Input consists of an integer.  
 Output consists of an integer.  
    
 Sample Input :  
 9999  
 Sample Output :  
 9

**ANSWER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

class UserProgramCode

{

public static int getDigitSum(int input)

{

int sum = 0;

int sum1 = 0;

int i = 0;

int j = 0;

while (input > 0)

{

i = (input % 10);

sum = sum + i;

input = input / 10;

}

if (sum < 10)

{

return sum;

}

else

{

while (sum > 0)

{

j = (sum % 10);

sum1 = sum1 + j;

sum = sum / 10;

}

}

return sum1;

}

}

class Program

{

public static void Main(string[] args)

{

int input, output;

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

output = UserProgramCode.getDigitSum(input);

Console.WriteLine(output);

Console.ReadLine();

}

}

**23. EMI Calculation**

**QUESTION:**

EMI Calculation  
 A person wants to apply for loan, but bank wants to check whether the person is current employee or retired or student based on given Date Of Birth (input1) and Loan EMI (input2). 1) The Bank can approve loan for student of Rs 200000 with rate of interest of 3% per Annum if age <= 22. 2) The Bank can approve loan for employee of Rs 300000 with rate of interest of 5% per Annum if age>22 and age=<45. 3) The Bank can approve loan for retired of Rs 500000 with rate of interest of 7% per Annum if age>45 and age<=100. Write a program to calculate the EMI . Note :   
 i) The Loan Amount has to be cleared in either 12/24/36/48 EMI months which is provided as input2. ii) Round the output to the nearest integer if required. iii)Age calculation to be done with respect to current date. Business Rules: i) If Date of Birth is not given in proper fomat(dd-MM-yyyy), then print -1. ii) If EMI month is not 12 , 24 , 36 or 48 , then print -2.    
 Create a class named UserProgramCode that has the following static method   
 public static int checkEmpAgeEligible(String input1, int input2)  
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 Input consists of a string that corresponds to Date of Birth and an integer that corresponds to EMI months.  
 Refer business rules and sample output for formatting specifications.  
 Sample Input 1 :  
 01-11-1983 12    
 Sample Output 1 :  
 26250 Sample Input 2 : 01/11/1983 48    
 Sample Output 2 :  
 -1 Sample Input 3 :  
 01-11-1983 48    
 Sample Output 3 :  
 7500

**ANSWER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace EMI

{

class Program

{

static void Main(string[] args)

{

string dob = Console.ReadLine();

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

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

{

Console.WriteLine(op);

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Xml;

using System.Text.RegularExpressions;

namespace EMI

{

class UserProgramCode

{

public static int inst(string dob, int month)

{

if ((month != 12) && (month != 24) && (month != 36) && (month != 48))

{

return -2;

}

DateTime dt1;

int inst = 0;

double amount1;

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

if (i == true)

{

int year = DateTime.Now.Year - dt1.Year;

int mont = DateTime.Now.Month - dt1.Month;

string maxdate = "2022-06-06";

DateTime dt2 = Convert.ToDateTime(maxdate);

DateTime dt3 = DateTime.Now.Date;

DateTime dt4 = dt3.AddMonths(month);

if (dt4 > dt2)

{

return -4;

}

if (mont < 0)

{

year = year - 1;

mont = mont + 12;

}

if (year <= 22)

{

amount1 = (double)200000 \* 1.03;

inst = (int)amount1 / month;

}

if (year > 22 && year <= 45)

{

amount1 = (double)300000 \* 1.05;

inst = (int)amount1 / month;

}

if (year > 45 && year <= 100)

{

amount1 = (double)500000 \* 1.07;

inst = (int)amount1 / month;

}

return inst;

}

else

return -1;

}

}

}

**24. validatePassword**

**QUESTION:**

Validate Password Write a method to validate given password. Apply following validations: 1. Minimum length should be 8 characters 2. Must contain any one of these three special characters @ or \_ or # 3. May contain numbers or alphabets. 4. Should not start with special character or number 5. Should not end with special character Include a UserProgramCode with a static method validatePassword.The method must return an integer 1 or -1. if it returns 1 then print a message "Valid Password". If the method returns -1 then print a message "Invalid Password". Create a class Program which gets a string as input and calls the static method validatePassword present in the UserProgramCode. Input and Output Format: Input is a string which is the password. Output is also a string which prints a message "Valid Password" or "Invalid Password". Sample Input 1: #bzdfh123c Sample Output 1: Invalid Password Sample Input 2 : jgu\_123dfsd3 Sample Output 2: Valid Password

**ANSWER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

class Program

{

public static void Main( string[] args )

{

String password=Console.ReadLine();

int result=UserProgramCode.validatePassword(password);

if(result==1)

Console.WriteLine("Valid password");

else

Console.WriteLine("Invalid password");

Console.ReadKey();

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

class UserProgramCode

{

public static int validatePassword(String input)

{

int output1 = 0;

if (input.Length >= 8)

{

char[] ch = input.ToCharArray();

if (char.IsLetter(ch.ElementAt(0)) && char.IsLetterOrDigit(ch.ElementAt(ch.Length - 1)))

{

int splchar = 0;

for (int i = 0; i < ch.Length - 2; i++)

{

if (char.IsLetterOrDigit(ch.ElementAt(i)))

{

}

else if (ch.ElementAt(i) == '#' || ch.ElementAt(i) == '\_' || ch.ElementAt(i) == '@')

splchar++;

}

if (splchar >= 1)

{

output1 = 1;

}

}

else

output1 = -1;

}

return output1;

}

}

**25. Reverse Number**

**QUESTION:**

Reverse Number  
 Write a program to read a positive number as input and to get the reverse of the given number and print it as output.   
    
 Example:  
 input = 543  
 output = 345  
    
 Include a class UserProgramCode with a static method reverseNumber which accepts an Integer. The return type (Integer) should return the reverse of the given input.  
 Create a Class Program which would be used to accept an Integer, and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of an Integer.  
 Output consists of an Integer, the reverse of the given Input.  
    
 Refer sample output for formatting specifications.  
    
 Sample Input 1:  
 543  
 Sample Output 1:  
 345

**ANSWER:**

question 39 :

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace question36

{

class Program

{

static void Main(string[] args)

{

int n, c=0;

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

if (n > 0)

{

c = UserProgramCode.reverseNumber(n);

}

Console.WriteLine(c);

}

}

class UserProgramCode

{

public static int reverseNumber(int a)

{

int rem, sum = 0;

while (a > 0)

{

rem = a % 10;

sum = (sum\*10) + rem;

a = a / 10;

}

return (sum);

}

}

}

**26.** **Reverse and Format**

**QUESTION:**

Reverse and Format  
 Write a program to read a String and a character and to reverse the string and return it in a format such that each character is separated by the given character. Print the final string.  
    
 Example:  
 input1: "Rabbit"  
 input2: '-'  
 output: "t-i-b-b-a-R"  
    
 Include a class UserProgramCode with a static method reshape which accepts a string and a character. The return type (String) should return the final String.  
 Create a Class Program which would be used to read a string and a character and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of a string and a character.  
 Output consists of a String(the final output).  
 Refer sample output for formatting specifications.  
    
 Sample Input:  
 Rabbit  
 -  
 Sample Output:  
 t-i-b-b-a-R

**ANSWER:**

using System;

class Program

{

public static void Main( string[] args )

{

string str=Console.ReadLine();

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

Console.WriteLine(UserProgramCode.reshape(str,ch));

Console.Read();

}

}

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

USER PROGRAM

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

using System;

class UserProgramCode

{

public static string reshape(string str, char ch)

{

int l = str.Length;

string sree = "";

char[] temp = str.ToCharArray();

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

{

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

if (i != 0)

{

sree = string.Concat(sree, ch);

}

}

return (sree);

}

}

**27. SumOfSquaresOfDigits**

**QUESTION:**

Sum of Squares of Digits  
 Write a program to find out the sum of squares of digits in a given number.  
    
 Example:  
 input = 321  
 output = (3\*3+2\*2+1\*1) = 14   
 Include a class UserProgramCode with a static method getSumOfSquaresOfDigits that accepts an integer and returns an integer.  
    
 Create a class Program which would get the input and call the static method getSumOfSquaresOfDigits present in the UserProgramCode.  
 Input and Output Format: Input consists of an Integer. Output consists of an integer.  
 Sample Input 1: 123 Sample Output 1: 14 Sample Input 2: 102 Sample Output 2: 5

**ANSWER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

class Program

{

static void Main(string[] args)

{

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

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

Console.ReadKey();

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

class Program

{

static void Main(string[] args)

{

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

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

Console.ReadKey();

}

}

class UserProgramCode

{

public static int getSumOfSquaresOfDigits(int n)

{

// fill your code here

int sum = 0, squareDigit, digit, num;

num = n;

while (num != 0)

{

digit = num % 10;

squareDigit = digit \* digit;

sum = sum + squareDigit;

num = num / 10;

}

return sum;

}

}

**28. Electricity Bill**

**QUESTION:**

Calculate Bill  
    
 Write a program to calculate the bill given the previous reading , current reading and per unit charge as inputs.    
 Example:  
 input1 =ABC2012345  
 input2 = ABC2012660  
 input3 = 4  
 Bill = (12660 - 12345 ) \* 4  
 output = 1260  
 Include a class UserProgramCode with static method calculateBill() that accepts 2 strings corresponding to the previous reading and current reading and an integer that corresponds to the per unit charge. This method returns an integer that corresponds to the bill amount to be paid. Create a class Program which would get the inputs and call the static method calculateBill() present in the UserProgramCode.  
    
 Input and Output Format:  
 Reading Format - XXXXXAAAAA where XXXXX is consumer number and AAAAA is meter reading.  
 Input1 is a String - previous reading of the consumer  
 Input2 is a String - current reading of the consumer Input3 is an integer - per unit charge to the consumer output is an integer - Calculated BILL value.    
 Metric BILL Formula: Bill=(current reading-previous reading)\*per unit charge  
    
 Sample Input 1: ABC2012345 ABC2012660 4 Sample Output 1: 1260  
    
    
  

**ANSWER:**

qn..29

class Program

{

static void Main(string[] args)

{

string s1 = Console.ReadLine();

string s2 = Console.ReadLine();

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

int sl = UserProgramCode.calculateBill(s1,s2,n);

Console.WriteLine(sl);

Console.ReadLine();

}

}

class UserProgramCode

{

public static int calculateBill(string s1,string s2,int n)

{

int sum = 0;

string ss1 = s1.Substring(5);

string ss2 = s2.Substring(5);

int a = int.Parse(ss1);

int b = int.Parse(ss2);

sum = (b - a) \* n;

return sum;

}

}

**29. Check Batch Code**

**QUESTION:**

Check Batch Code  
 Write a program which will check if the given input string follows the below format and print the output according to the conditions given below. 1. The format of the string should be 'AAABBCCXXX' where a. AAA represents the location of the batch CHN -- Chennai CBE -- Coimbatore KOC - Kochi PUN - Pune BGL - Bangalore HYD - Hyderabad KOL - Kolkata Business rules: THe characters 'AAA' should not be other than the above specified values(Only Capitals). If it is other than these characters, print -1. b. BB and XXX in the format represents numerals between 0-9. BB Represents the year and XXX represents the batch code.If other than these are present print -2. c.CC in the format should be only 'DN', if not print -3. All the characters in the input string are in upper case. Please make sure you dont do a spell mistake in the output string. Example 1: Input : CHN13DN014 The output string should be in the following format. DotNet batch 014 has joined in 2013 year and is at Chennai Location    
 Create a class named UserProgramCode that has the following static method   
 public static string checkBatch(string input1)  
    
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 Input consists of a string.  
 Refer business rules and sample output for formatting specifications.    
 Sample Input 1 :  
 CHN13DN014    
 Sample Output 1 :  
 DotNet batch 014 has joined in 2013 year and is at Chennai Location Sample Input 2 :  
 PUN13DN004  
    
 Sample Output 2 : DotNet batch 004 has joined in 2013 year and is at Pune Location Sample Input 3 :  
 BGL14DN014    
 Sample Output 3 :  
 DotNet batch 014 has joined in 2014 year and is at Bangalore Location

**ANSWER:**

5.STRING BATCH CODE:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication90

{

class Program

{

static void Main(string[] args)

{

string s = Console.ReadLine();

var x = Class1.batch(s);

Console.WriteLine(x);

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

namespace ConsoleApplication90

{

class Class1

{

public static string batch(string s)

{

string op = "";

Regex re = new Regex(@"^([A-Z]{3}[0-9]{2}[D][N][0-9]{3})$");

if (re.IsMatch(s))

{

string loc = s.Substring(0, 3);

string output = "";

switch (loc)

{

case "CHN":

output = "Chennai";

break;

case "CBE":

output = "Coimbatore";

break;

case "KOC":

output = "Kochi";

break;

case "PUN":

output = "PUNE";

break;

case "BGL":

output = "BANGALORE";

break;

case "HYD":

output = "HYDERABAD";

break;

case "KOL":

output = "KOLKATTA";

break;

default:

output = "-1";

break;

}

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

}

else

{

op = "-4";

}

return op;

}

}

}

**30. Relative Order**

**QUESTION:**

Relative Order  
    
 Given two input integer arrays input1 and input2, write a program to sort input1 in such a way that the relative order among the elements will be same as those are in input2. For the elements not present in input2,append them at last in sorted order. Business Rules : 1. If any of the given inputs contains any negative number, then print -1. 2. If any of the elements in input 2 array is not available in input 1 array, then print -2. 3. If there are less than 3 elements or more than 15 elements in the input1 array, print -3.    
 Example  
 Input Array 1 = {2,1,2,5,7,1,9,3,6,8,8} Input Array 2 = {2,1,8,3| Output Array = {2,2,1,1,8,8,3,5,6,7,9} Create a class named UserProgramCode that has the following static method   
 public static int[] relativeOrder(int[] input1,int[] input2) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 The first line of the input consists of an integer, n that corresponds to the number of elements in the input array 1.  
 The next 'n' lines of input consist of elements in the input array 1.  
 The next line of the input consists of an integer, m that corresponds to the number of elements in the input array 2.  
 The next 'm' lines of input consist of elements in the input array 1.  
 Refer business rules and sample output for formatting specifications. Sample Input 1 : 11 2 1 2 5 7 1 9 3 6 8 8 4 2 1 8 3 Sample Output 1 : 2 2 1 1 8 8 3 5 6 7 9    
 Sample Input 2 : 8 2 1 5 7 9 3 6 8 4 2 1 8 3    
 Sample Output 2 : 2 1 8 3 5 6 7 9 Sample Input 3 : 11 2 1 2 -5 7 1 9 3 6 8 8 4 2 1 8 3    
 Sample Output 3 :  
 -1

**ANSWER:**

program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Relative\_Order

{

class Program

{

static void Main(string[] args)

{

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

int[] arr1 = new int[n1];

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

{

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

}

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

int[] arr2 = new int[n2];

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

{

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

}

int[] op = UserProgramCode.Relative\_order(arr1, arr2);

{

foreach (var item in op)

{

Console.WriteLine(item);

}

}

}

}

}

userprogram.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Relative\_Order

{

class UserProgramCode

{

public static int[] Relative\_order(int[] arr1, int[] arr2)

{

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

List<int> list1 = new List<int>();

for (int i = 0; i < arr2.Length; i++)

{

for (int j = 0; j < arr1.Length; j++)

{

if (arr1[j] == arr2[i])

{

list.Add(arr1[j]);

}

}

}

for (int i = 0; i < arr1.Length; i++)

{

if(!list.Contains(arr1[i]))

{

list1.Add(arr1[i]);

}

}

list1.Sort();

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

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

{

finalList.Add(list1[i]);

}

return finalList.ToArray();

}

}

}