**61.Find Average**  
 Write a program to read an Integer (the size of the List) and the List of Integers and find the average of the numbers as a float value. Print the average.  
    
 Print Error Code “Negative numbers present” when inputs other than positive numbers is given.  
    
 Include a class UserProgramCode with a static method findAverage which accepts an Integer list. The return type (Float) should return the average value. If negative numbers are present in the array, then return -1.  
 Create a Class Program which would be used to accept an Integer and an Integer list, and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of n+1 Integers, where the first number corresponds the size of the array, followed by the array elements.  
 Output consists of a Float, the average value, or a String “Negative numbers present” if a negative number is present in the array.  
    
 Refer sample output for formatting specifications.  
    
 Sample Input 1:  
 4  
 2  
 3  
 2  
 3  
 Sample Output 1:  
 2.5  
    
 Sample Input 2:  
 2  
 1  
 -2  
 Sample Output 2:  
 Negative numbers present

46.find average:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication18

{

class UserProgramCode

{

public static float compute(int[] array, int size)

{

float avg,sum = 0;

int i;

foreach (int a in array)

{

if (a < 0)

return -1;

}

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

{

sum = sum + array[i];

}

avg = sum / size;

return avg;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication18

{

class Program

{

static void Main(string[] args)

{

UserProgramCode u = new UserProgramCode();

int n;

float avg;

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

int[] a = new int[n];

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

{

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

}

avg = UserProgramCode.compute(a, n);

if (avg == -1)

{

Console.WriteLine("Negative numbers present");

}

else

Console.WriteLine(String.Format("{0:0.0}",avg));

}

}

}

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

Sample Input 1: qwerty

Sample Output 1: 6

Sample Input 2: 12345

Sample Output 2: 5

Count Characters:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication18

{

class UserProgramCode

{

public static int countcharachters(string[] s)

{

int sum = 0,flag = 0 ;

foreach (string s1 in s)

{

char[] ch = s1.ToCharArray();

foreach (char c in ch)

{

if (char.IsLetter(c))

{

flag++;

}

}

}

foreach (string s1 in s)

{

sum = sum + s1.Length;

}

if(flag==sum)

return sum;

else

return -1;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication18

{

class Program

{

static void Main(string[] args)

{

UserProgramCode u=new UserProgramCode();

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

string[] s=new string[n];

int result;

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

s[i] = Console.ReadLine();

result=UserProgramCode.countcharachters(s);

if(result==-1)

Console.WriteLine("Invalid Input");

else

Console.WriteLine(result);

}

}

}

}

63.Length of the longest string  
    
 Write code to find the length of the longest string in the given string list.  
 Include a class UserProgramCode with static method longestWordLength that accepts the String list and the return type should be int Create a class Program which would get the input and call the static method longestWordLength(String[] array) present in the UserProgramCode. The longestWordLength(String[] array) returns the length of the longest string    
 Input and Output Format:  
 The first integer corresponds to n, the number of elements in the list. The next 'n' integers correspond to the elements in the String list. SAMPLE INPUT 1  
 2   
 Black   
 Blue   
    
 SAMPLE OUTPUT 1  
 5

Length of the longest word:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Fwd\_Prgs

{

public class UserProgramCode

{

public static string longestWordLength(string[] s)

{

int sum = 0;

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

{

if (s[i].Length > sum)

{

sum = s[i].Length;

}

}

return sum.ToString();

}

}

class Program

{

static void Main(string[] args)

{

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

string[] str=new string[n];

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

{

str[i] = Console.ReadLine();

}

string res = UserProgramCode.longestWordLength(str);

Console.WriteLine(res);

}

}

}

64. Three Digits  
 Write a program to read a string and check if the given string is in the format "CTS-XXX" where XXX is a three digit number.  
 Include a class UserProgramCode with a static method validatestrings which accepts a string and returns an integer.. The function returns 1 if the string format is correct,  else returns -1.   
 Create a Class Program which would be used to accept a String and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of a string.   
 Output consists of a string (Valid or Invalid).  
 Refer sample output for formatting specifications.  
 Sample Input 1:  
 CTS-215  
 Sample Output 1:  
 Valid  
    
 Sample Input 2:  
 CTS-2L5  
 Sample Output 2:  
 Invalid

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace level1\_60

{

class Program

{

static void Main(string[] args)

{

int x;

string s = Console.ReadLine();

x = UserProgramCode.validatestrings(s);

if (x == 1)

{

Console.WriteLine("valid");

}

else

{

Console.WriteLine("Invalid");

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

namespace level1\_60

{

class UserProgramCode

{

public static int validatestrings(string str)

{

int output1 = 0;

Regex reg = new Regex(@"^([C]+[T]+[S]+[-]+([0-9]{3}))$");

if (reg.IsMatch(str))

{

output1 = 1;

}

else

{

output1 = -1;

}

return output1;

}

}

}

65. All Vowels  
    
 Write a Program to check if given word contains exactly five vowels and the vowels are in alphabetical order. Assume there is no repetition of any vowel in the given string and all letters are in lower case.   
 Include a class UserProgramCode with a static method testOrderVowels which accepts a string and returns an integer.  The method returns 1 if the condition  stated above is satisfied. Else the method returns -1.  
 Create a Class Program which would be used to read a String and call the static method present in UserProgramCode. If the method returns 1, print 'valid'. Else print 'invalid'.  
 Input and Output Format:  
 Input consists of a string with maximum size of 100 characters.   
 Output consists of a single string.  
 Refer sample output for formatting specifications.  
 Sample Input 1:  
 acebisouzz  
 Sample Output 1:  
 valid    
 Sample Input 2:  
 alphabet Sample Output 2:  
 invalid

ALL VOWELS.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

namespace ConsoleApplication20

{

class Program

{

static void Main(string[] args)

{

string input1;

int output1;

input1 = Console.ReadLine();

output1 = UserProgramCode.testOrderVowels(input1);

if (output1.Equals(1))

{

Console.WriteLine("valid");

}

else

{

Console.WriteLine("invalid");

}

Console.Read();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication20

{

class UserProgramCode

{

public static int testOrderVowels(string input1)

{

int output1 = 0;

StringBuilder sb = new StringBuilder();

char[] ch = input1.ToCharArray();

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

{

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

{

sb.Append(ch[i]);

}

if (sb.ToString() == "aeiou")

{

output1 = 1;

}

else

{

output1 = -1;

}

}

return output1;

}

}

}

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

String reversal

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

string s = Console.ReadLine();

string a = UserProgramCode.reverseString(s);

if (a == "-1")

Console.WriteLine("Invalid Input");

else

Console.WriteLine(a);

Console.ReadLine();

}

}

class UserProgramCode

{

public static string reverseString(string a)

{

int l=a.Length;

if (a.Any(ch => !(Char.IsLetterOrDigit(ch) || char.IsWhiteSpace(ch))))

return "-1";

StringBuilder sb = new StringBuilder();

char[] c;

string[] s;

s=a.Split(' ');

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

{

c= s[i].ToCharArray();

Array.Reverse(c);

sb.Append(c);

sb.Append(" ");

}

return sb.ToString();

}

}

}

67. Finding common Elements in multiples of 3  
 Write a program to find the common elements from all the three input integer lists which are also multiple of 3. Sort the result in descending order and print it. Include a class UserProgramCode with static method  FindCommonElements() which accepts 3 integer List. The return type is List<int> which returns common elements in the List.  
    
 Example : Input :  
 3 ----------------------------------------------------->First list size 21 6 8 2------------------------------------------------------>Second list size 21 6 4------------------------------------------------------>Third list size 4 89 21 6 Output : 21 6  
 Create a class Program which would get the input and call the static method FindCommonElements() present in the UserProgramCode.  
    
 Input and Output Format: Refer Example for input Format. Output is a list which contains the common elements in multiples of 3 or a string as specified below.  
 In FindCommonElements()  
 If there is no common elements found in all of three input lists then assign 0 to the first element of the output list and return the list. If any of the input lists have negative element then then assign -1 to the first element of the output list and return the list. If any of the input lists have element value greater than 500 then assign -2 to the first element of the output list and return the list. In Program class    If the method returns a list with the first element being 0, then print "No match found".    If the method returns a list with the first element being -1, then print  "The list contains negative values".    If the method returns a list with the first element being -2, then print  "The elements of the list should be less than or equal to 500".    Otherwise print the result.      
 SAMPLE INPUT 1:  
 3 21 6 8 2 21 9 4 4 89 21 56 SAMPLE OUTPUT 1: 21  
 SAMPLE INPUT 2:  
 4 13 -27 44 9 5 24 9 41 56 8 4 27 24 -9 8 SAMPLE OUTPUT 2: The list contains negative values  
 SAMPLE INPUT 3:  
 3 33 27 444 5 24 9 41 56 8 4 27 24 78 55 SAMPLE OUTPUT 3: No match found  
 SAMPLE INPUT 4:  
 8 3 666 7 4 9 24 21 8 7 16 17 24 9 21 56 8 6 3 6 7 24 9 8 SAMPLE OUTPUT 4: The elements of the list should be less than or equal to 500

67.Find common elements

FIndCommon

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace fFindCommon

{

class UserProgramCode

{

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

{

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

int flag = 0;

int flag1 = 0;

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

{

if (list1[i] < 0)

{

final.Add(-1);

flag = 1;

}

}

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

{

if (list2[i] < 0)

{

final.Add(-2);

flag1 = 2;

}

}

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

{

final.Clear();

final.Add(-3);

return final;

}

else if (flag == 1)

return final;

else if(flag1==2)

return final;

else

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

{

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

{

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

{

final.Add(list1[i]);

}

}

}

final.Sort();

return final;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace fFindCommon

{

class Program

{

static void Main(string[] args)

{

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

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

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

{

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

}

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

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

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

{

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

}

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

foreach (int item in op)

{

Console.WriteLine(item);

}

Console.ReadLine();

}

}

}

68. MaxMin Sum  
    
 Write a program that accepts 3 integer inputs and finds the sum of maximum and minimum. Business Rules : 1) If any/ or all of the input value is negative then print -1. 2) If any two or all the values in the Input are same then print -2. Example 1: Input1: 25 Input2: 2 Input3: 95 Output : 97 (Min 2 + Max 95) Example 2: Input1: -15 Input2: 49 Input3: 5 Output : -1    
 Create a class named UserProgramCode that has the following static method   
 public static int sumMaxMin(int input1, int input2, int input3)  
    
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.  
 Input and Output Format: Input consists of 3 integers.  
 Output is an integer. Refer sample output and business rules Sample Input 1: 25  
 2  
 95  
    
 Sample Output 1:  
 97  
    
 Sample Input 2: -15  
 49  
 5  
    
 Sample Output 2:  
 -1

68.Maxminsum

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication13

{

class userprogramcode

{

public static int sumMaxMin(int ip1, int ip2, int ip3)

{

int ans,a,b;

int[] t1 = new int[3];

t1[0] = ip1;

t1[1] = ip2;

t1[2] = ip3;

for (int i = 0; i < 3; i++)

if (t1[i] < 0)

return -1;

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

{

for (int j = i + 1; j < 3; j++)

{

if (t1[i] == t1[j])

return -2;

}

}

a = t1.Max();

b = t1.Min();

ans = a + b;

return ans;

}

}

class Program

{

static void Main(string[] args)

{

int x,y,z,k;

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

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

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

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

k =userprogramcode.sumMaxMin(x,y,z);

Console.WriteLine(k);

}

}

}

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

69.List the elements

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace levelI04

{

class Program

{

static void Main(string[] args)

{

int n,i;

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

int[] arr = new int[n];

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

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

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

int[] ans = UserProgramCode.GetElements(arr, limit);

if(ans[0]==-1)

Console.WriteLine("No element found");

else

{

foreach (int item in ans)

{

Console.WriteLine(item);

}

}

}

}

class UserProgramCode

{

public static int[] GetElements(int[] arr,int limit)

{

int n = arr.Length;

int[] temp=new int[n];

int i,j;

i=0;

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

{

if(arr[j]>limit)

{

temp[i]=arr[j];

i++;

}

}

if (temp[0] == 0)

{

temp[0] = -1;

return temp;

}

else

{

Console.WriteLine("");

Array.Sort(temp);

Array.Reverse(temp);

return temp;

}

}

}

}

70. Reimbursement  
 Hina University offers fees reimbursement scheme to its students according to the percentage they secure in board examinations, as per the below criteria.  
    
 Category A: For 80% to 85% (inclusive of border values) secured in exam: Refundable amount is 40% of the fees paid by the student during the start of the academic year and a cash award.  
    
 Category B: For 86% to 90% (inclusive of border values) secured in exam: Refundable amount is 50% of the fees paid by the student during the start of the academic year and a cash award.  
    
 Category C: For above 90% : Refundable amount is 60% of the fees paid by the student during the start of the academic year and a cash award.  
    
 The University also awards the students with a cash prize of Rs. 3000, Rs. 5000 and Rs. 7000 for Categories A, B and C respectively.  
 However, for the student to be eligible for reimbursement, he should NOT have a backlog (arrear) in any subject.  
    
 Write a program that calculates the total amount the student receives from the University which includes refundable fee amount and cash prize, according to:   
    
 1. The fees he pays at the start of the academic year, (this is the first input).  
 2. His/Her percentage of marks in the exams and (this is the second input).  
 3. His backlog status(Boolean : True if there is arrear and False if there is no arrear, this is the third input).  
 (Total amount the student receives = Refundable fee amount + Cash Prize)  
    
 Validation Rules :  
    
 1. Only positive number greater than 25000 should be given for fees amount else return -1.  
 2. Only numbers in the range 80 to 100 should be for percentage else return -2.  
    
 Include a class UserProgramCode with a static method calulateAmountRefundable which accept the fees, mark percentage and the backlog status. The return type (integer) should return the Refundable amount, or -1, or -2, accordingly.  
 Create a Class Program which would be used to accept two integers, and a boolean value, and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of double ,integer and a boolean value, where double corresponds to the fees, integer corresponds to the percentage and the boolean values corresponds to the backlog status.  
 Output consists of an Integer or one of the 2 strings ("Low fees amount" or "Invalid percentage").  
    
 Refer sample output for formatting specifications.  
    
 Sample Input 1:  
 25000  
 82  
 false  
 Sample Output 1:  
 13000  
    
 Sample Input 2:  
 20000  
 82  
 false  
 Sample Output 2:  
 Low fees amount  
    
 Sample Input 3:  
 20000  
 72  
 false  
 Sample Output 3:  
 Invalid percentage

70.Reimburesment.

namespace SM1

{

class UserProgramCode

{

public static int calulateAmountRefundable(double fee, int marks, bool arr)

{

double total;

if (marks < 80 || marks > 100)

return -2;

else if (fee < 25000)

return -1;

else

{

if (!arr)

{

if (marks >= 80 && marks <= 85)

{

total = (40 \* fee) / 100;

total = total + 3000;

return (int)total;

}

else if (marks >= 86 && marks <= 90)

{

total = (50 \* fee) / 100;

total = total + 5000;

return (int)total;

}

else if (marks >= 90 && marks <= 100)

{

total = (60 \* fee) / 100;

total = total + 7000;

return (int)total;

}

else

return 0;

}

else

{

return -3;

}

}

}

}

class Program

{

static void Main(string[] args)

{

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

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

bool arrer = (bool)Convert.ToBoolean(Console.ReadLine());

int res = UserProgramCode.calulateAmountRefundable(fees, mar, arrer);

if(res==-1)

Console.WriteLine("Low fees amount");

else if(res==-2)

Console.WriteLine("Invalid percentage");

else

Console.WriteLine(res);

Console.ReadLine();

}

}

}

71. Next Highest Number  
 Write a progam that accepts an integer input and finds out all the combinations of the numbers possible with all the digits present in the input integer and then from the list of combinations,picks up the next higher number than the given input and prints it. Business Rules : 1. If the given input integer is a negative number, then print -1 . 2. If the input contains more than 3 digits,then print -2. 3. If any of the digits present in input are found repetitive, then print -3 . Create a class named UserProgramCode that has the following static method  
 public static int nextHighestNumber(int input1)  
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:  
 Input consists of an integer.  
 Output consists of an integer.  
    
 Sample Input 1:  
 376 Sample Output 1:  
 637 Sample Input 2: -236  
 Sample Output 2: -1

NEXTHIGHESTNUMBER

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

namespace ConsoleApplication20

{

class Program

{

static void Main(string[] args)

{

int n,output;

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

output = UserProgramCode.nextHighestNumber(n);

Console.WriteLine(output);

Console.Read();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication20

{

class UserProgramCode

{

public static int nextHighestNumber(int n)

{

int num = n;

int l = n.ToString().Length;

int next;

if (l == 1)

{

return num;

}

else if (num < 0)

{

return -1;

}

else if (l > 3)

{

return -2;

}

else

{

if (l == 2)

{

int rem1 = n % 10;

n = n / 10;

next = (rem1 \* 10) + n;

return next;

}

else

{

int rem1 = n % 10;

n = n / 10;

int rem2 = n % 10;

n = n / 10;

if (rem1 > rem2)

{

next = (n \* 100) + (rem1 \* 10) + rem2;

return next;

}

else if (n < rem1 && rem2 != rem1)

{

next = (rem1 \* 100) + (n \* 10) + rem2;

return next;

}

else if (n < rem2 && rem2 != rem1)

{

next = (rem2 \* 100) + (n) + rem1 \* 10;

return next;

}

else

{

return -3;

}

}

}

}

}

}

72.Sum Non Prime Numbers  
    
 Write a program to calculate the sum of all the non prime positive numbers less than or equal to the given number.  
    
 Note: prime is a natural number greater than 1 that has no positive divisors other than 1 and itself  
    
 Example:  
 input = 9  
 Prime numbers = 2,3,5 and 7  
 output = 1+4+6+8+9=28  
    
 Include a class UserProgramCode with a static method “addNumbers” that accepts an integer arguement and returns an integer.  
    
 Create a class Program which would get an integer as input and call the static method addNumbers present in the UserProgramCode.  
    
 Input and Output Format:  
 Input consists of an integer.  
 Output consists of an integer.  
    
 Sample Input:  
 9  
    
 Sample Output:  
 28

SumNonPrimeNumbers

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace sum\_non\_prime

{

class Program

{

static void Main(string[] args)

{

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

int op = UserProgramCode.nonprime(num);

Console.WriteLine(op);

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace sum\_non\_prime

{

class UserProgramCode

{

public static int nonprime(int num)

{

int sum = 1;

for (int i = 2; i <=num; i++)

{

int c = 0;

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

{

if (i % j == 0)

{

c++;

}

}

if (c != 2)

{

sum = sum + i;

}

}

return sum;

}

}

}

73. Berth Type  
 Ram books 3 train tickets for his father,grandfather and himself.Their seat numbers are given as input1,input2 and input3 respectively. Each person is assigned a seat based on the berth type preference(Lower [L], Middle [M], Upper [U], SideLower [SL] and SideUpper[SU]). The berth type can be identified based on the following steps: 1. Divide the seat number by 8 2. If the remainder is either 1 or 4, then the berth type is Lower If the remainder is either 2 or 5, then the berth type is Middle If the remainder is either 3 or 6, then the berth type is Upper If the remainder is 7, then the berth type is SideLower If the remainder is 0, then the berth type is SideUpper 3. Based on the grandfather's berth type, print the following: If berth type is Lower(L), print  
 Lower berth provided as per request If berth type is other than Lower(L),swap it with other seat numbers which is lower and print  
 Your seat has been swapped from XX to YY as per preference request If berth type is not Lower(L) for all the three seats,print  
 Your seat will be changed on the date of travel here XX is the initial seat number of Ram's grandfather and YY is the swapped seat number.First try swapping with Ram's seat if found lower,else with Ram's fathers seat.Maximum seat number limit - 1000. Business rules: 1.If any of the seat number is zero or negative , print “Invalid Seat Number”. 2.If any of the seat numbers contain any alphabets or special characters , print “Invalid Input” Create a class named UserProgramCode that has the following static method   
 public static string checkBerthType(string input1,string input2,string input3)  
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 Input consists of 3 strings – input1 (Ram's father's berth number), input2 (Ram's grandfather's berth number) and input3 (Ram's berth number).  
 Output consists of a string.  
 Refer business rules and sample output for formatting specifications.    
 Sample Input 1 : 76 75 78  
 Sample Output 1 :   
 Your seat has been swapped from 75 to 76 as per preference request Sample Input 2 :  
 76 -75 78  
 Sample Output 2 :  
 Invalid Seat Number

BERTH TYPE

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace BerthType

{

class Program

{

static void Main(string[] args)

{

String s1 = Console.ReadLine();

String s2 = Console.ReadLine();

String s3 = Console.ReadLine();

String r = UserProgramCode.Berth\_type(s1, s2, s3);

Console.WriteLine(r);

Console.ReadKey();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace BerthType

{

class UserProgramCode

{

public static String Berth\_type(String s1, String s2, String s3)

{

String b1 = "", b2 = "", b3 = "";

int m1, m2, m3, temp = 0;

String res = "";

try

{

m1 = Convert.ToInt32(s1);

m2 = Convert.ToInt32(s2);

m3 = Convert.ToInt32(s3);

if (m1 < 0 || m2 < 0 || m3 < 0)

{

throw new Exception();

}

}

catch

{

res = "Invalid Seat No.";

return res;

}

int r1 = m1 % 8;

int r2 = m2 % 8;

int r3 = m3 % 8;

if ((r1 == 1 || r1 == 4))

{

b1 = "Lower";

}

if ((r2 == 1 || r2 == 4))

{

b2 = "Lower";

}

if ((r3 == 1 || r3 == 4))

{

b3 = "Lower";

}

if (b2 == "Lower")

{

res = "Grandfather got Lower seat";

return res;

}

else if (b2 != "Lower")

{

if (b1 == "Lower")

{

res = "Your seat has been swapped from " + m2 + " to " + m1;

temp = m1;

m1 = m2;

m2 = temp;

return res;

}

else if (b3 == "Lower")

{

res = "Your seat has been swapped from " + m2 + " to " + m3;

temp = m3;

m3 = m2;

m2 = temp;

return res;

}

else

{

res = "Sorry your request can not be processed....";

return res;

}

}

return res;

}

}

}

74.BMI CALCULATOR

BMI Calculator Write a program to find the BMI of a person given their height(In Metres) and weight(In Kg) as inputs. Example : input1 = 70 input2 = 1.65 Metres BMI := 70/(1.65\*1.65) =25.711 output = Overweight Include a class UserProgramCode with static method BMICalc which accepts two float numbers. The return type is String. Create a class Program which would get the input and call the static method BMICalc present in the UserProgramCode. Input and Output Format: Input1 is a float - Weight(In Kg) Input2 is a float - Height (In Metres) Output is a string – Interpreted BMI value. Metric BMI Formula BMI = ( Weight in Kilograms / ( Height in Meters x Height in Meters ) ) Business rule: BMI Interpretation is given below Underweight = BMI of <18.5 Normalweight = BMI of 18.5–24.9 Overweight = BMI of 25–29.9 Obesity = BMI of 30 or greater If zero or negative number is given as input then return "InvalidInput" , otherwise return "Underweight","Normalweight", "Overweight","Obesity" as per Business rule. Sample Input 1: 70 1.65 Sample Output 1: Overweight Sample Input 2: 45 1 Sample Output 2: Obesity

.BMI calculator

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace levelI01

{

class Program

{

static void Main(string[] args)

{

float input1, input2;

input1 = float.Parse(Console.ReadLine());

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

Console.WriteLine(UserProgramCode.BMICalc(input1, input2));

}

}

class UserProgramCode

{

public static string BMICalc(float input1, float input2)

{

float bmi;

if (input1 <= 0 || input2 <= 0)

{

return("InvalidInput");

}

else

{

bmi = (input1 / (input2 \* input2));

if (bmi < 18.5)

return("Underweight");

else if (bmi >= 18.5 && bmi <= 24.9)

return("Normalweight");

else if (bmi >= 25 && bmi <= 29.9)

return("Overweight");

else if (bmi >= 30)

return("Obesity");

}

return ("null");

}

}

}

75. Validate Voter  
 Write a program to Validate the eligibility of the users for Voting in Election by accepting the input as Date Of Birth and the Date Of Election. Accept the dates as strings. Validate the DOB against the Date of Election, if it is atleast 18 yrs.   
    
 Business Rules:  
    
 1.Only date format “mm/dd/yyyy” should be given as input, if not return -1.  
    
 2. The eligible voting age is from 18 years.(Including 18)  
    
 3. If the age is valid for voting , then return 1.  
    
 4. If the age is invalid for voting , then return 0.  
    
 Include a class UserProgramCode with a static method validateVoter which accepts two Strings. The return type (Integer) should return a value according to the business rules.  
 Create a Class Program which would be used to accept two Strings, and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of two Strings, the first String corresponds to the DOB and the second String corresponds to the Date Of Election.  
 Output consists of a String, (“Invalid Date format” if -1 is returned, “Eligible” if 1 is returned, “Not Eligible” if 0 is returned.  
    
 Refer sample output for formatting specifications.  
    
 Sample Input 1:  
 12/29/1990  
 09/11/2014  
 Sample Output 1:  
 Eligible  
    
 Sample Input 2:  
 12/29/2010  
 09/11/2014  
 Sample Output 2:  
 Not Eligible  
    
 Sample Input 3:  
 32/29/1990  
 09/11/2014  
 Sample Output 3:  
 Invalid Date format

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ValidateVoter

{

class Program

{

static void Main(string[] args)

{

string dob = Console.ReadLine();

string doe = Console.ReadLine();

int op = UserProgramCode.ValidateVoter(dob, doe);

if(op ==1)

Console.WriteLine("Eligible");

else if(op==0)

Console.WriteLine("Not Eligible");

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ValidateVoter

{

class UserProgramCode

{

public static int ValidateVoter(string dob, string doe)

{

int output1 = 0;

DateTime dt;

DateTime dt1;

bool dobres = DateTime.TryParseExact(dob, "dd/mm/yyyy", null, System.Globalization.DateTimeStyles.None, out dt);

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

if (!(dobres == true && doeres==true))

{

int age = dt1.Year - dt.Year;

if (dt > dt1.AddYears(-age))

{

age--;

}

if (age >= 18)

{

output1 = 1;

}

else

{

output1 = 0;

}

}

return output1;

}

}

}

76. Class Division  
 Write a program to calculate the division/class obtained by the student when the marks obtained by a student in 5 different subjects are given as inputs.  
    
 The student gets a division/class as per the following rules:  
 Percentage above or equal to 60 - “First Class”.  
 Percentage between 50 and 59 - “Second Class”.  
 Percentage between 40 and 49 - “Third Class”.  
 Percentage less than 40 - “Failed”.  
    
 Include a class UserProgramCode with a static method calculateResult which accepts five integers. The return type (String) should return the class of the student.  
 Create a Class Program which would be used to accept 5 integer inputs and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of five integers.  
 Output consists of a String(class of the student).  
 Refer sample output for formatting specifications.  
    
 Sample Input 1:  
 41  
 45  
 46  
 40  
 41   
 Sample Output 1:  
 Third Class  
  

Class Division

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace levelI05

{

class Program

{

static void Main(string[] args)

{

int sub1, sub2, sub3, sub4, sub5;

sub1 = int.Parse(Console.ReadLine());

sub2 = int.Parse(Console.ReadLine());

sub3 = int.Parse(Console.ReadLine());

sub4 = int.Parse(Console.ReadLine());

sub5 = int.Parse(Console.ReadLine());

Console.WriteLine(UserProgramCode.calculateResult(sub1, sub2, sub3, sub4, sub5));

}

}

class UserProgramCode

{

public static string calculateResult(int sub1,int sub2,int sub3,int sub4,int sub5)

{

// int sub1,sub2,sub4,sub3,sub5;

int sum=sub1+sub2+sub3+sub4+sub5;

string str="";

int avg=sum/5;

if(avg>=60)

str="First Class";

else if(avg>=50 && avg<=59)

str="Second Class";

else if(avg>=40 && avg<=49)

str="Third Class";

else if(avg<40)

str="Failed";

return str;

}

}

}

77. Display Students Exam Eligibility status  
 A university has the following rules for a student to qualify for a degree with A as the main subject and B as the subsidiary subject:  
    
 Business Rule:  
    
 (a) He/She should get 55 percent or more(>=55%) in A and 45 percent or more(>=45%) in B.  
 (b) If he/she gets less than 55(<55%) percent in A he/she should get 55 percent or more(>=55%) in B. However, he/she should get at least 45 percent(>=45%) in A.  
 (c) If he/she gets less than 45 percent(<45%) in B and 65 percent or more(>=65%) in A he/she is allowed to reappear in an examination in B to qualify.  
 (d) In all other cases he/she is declared to have failed.  
    
 Write a code to display the student status according to above rules. Consider inputs as marks for both the subject.   
    
 Include a class UserProgramCode with static method FindResult() that accepts two integers. The return type should be String.  
    
 Create a class Program which would get the input and call the static method present in the UserProgramCode.  
    
 Input and Output Format :  
    
 Input1- % of Marks in A  
 Input2- % of Marks in B  
 Output1- Result( "P" for Pass ,"F" for fail ; "R" for Reappear)  
    
 If the input is more than 100 return "Invalid Input" from FindResult(int input1,int input2) else return appropriate result. Then display the result in Program class.    
 Sample Input 1: 42 45 Sample Output 1: F Sample Input 2: 105 05 Sample Output 2: Invalid Input

STUDENT EXAM ELIGIBILITY

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace StudentExamEligibility

{

class Program

{

static void Main(string[] args)

{

int a, b;

string result;

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

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

result = UserProgramCode.FindResult(a, b);

Console.WriteLine(result);

Console.ReadKey();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace StudentExamEligibility

{

class UserProgramCode

{

public static string FindResult(int a, int b)

{

string result;

if (a > 100 || b > 100)

{

result = "Invalid Input";

}

else

{

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

{

result = "P";

}

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

{

result = "P";

}

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

{

result = "R";

}

else

{

result = "F";

}

}

return result;

}

}

}

78.Colour Code  
 Write a program to find whther the given string corresponds to a valid colour code or not.  
 Write code to validate the given color code based on following rules:  
 - Must start with # symbol  
 - Must contain six characters after #  
 - It may contain alphabets from A-F (only upper case) or digits from 0-9    
 Example :  
 input = #FF9922  
 output = Valid  
    
 Include a class UserProgramCode with a static method validateColorCode. This method returns 1 if the input corresponds to a valid color code. Else this method returns -1.  
    
 Create a class Program which would get the input and call the static method validateColorCode present in the UserProgramCode.  
    
 Input and Output Format:  
 Input is a string - color code as value Output is a string - Valid or Invalid Sample Input 1: #FF9922 Sample Output 1: Valid Sample Input 2: 1234567 Sample Output 2: Invalid

class Program

{

static void Main(string[] args)

{

string str = Console.ReadLine();

int i = UserProgramCode.validateColorCode(str);

if (i == 1)

Console.WriteLine("Valid");

else

Console.WriteLine("Invalid");

Console.ReadLine();

}

}

class UserProgramCode

{

public static int validateColorCode(string s)

{

int flag = 0;

if(s.StartsWith("#"))

{

if (s.Length == 7)

{

char[] ch = s.ToCharArray();

for(int i=1;i<=6;i++)

{

if (char.IsDigit(ch[i]) || "ABCDEF".Contains(ch[i]))

{

flag = 1;

}

else

{

flag = 0;

break;

}

}

}

}

if (flag == 0)

return -1;

else

return 1;

}

}

79.   Reverse the adjacent pairs of letters  
 Write a program to swap the adjacent letters from the given string. If the string has an odd number of letters, the last letter is unchanged.  
    
 Include a class UserProgramCode with static method  swapPairs that accepts the string and return type should be string.  
 Create a class Program which would get the input and call the static method swapPairs present in the UserProgramCode.  
 If the input contains special characters or numbers, display "Invalid Input" in swapPairs()  otherwise display the resultant string.  
    
 Input Output format: The input and output should be a String. Sample input 1: Newyork Sample output 1: eNywrok Sample input 2: New!@ Sample output 2: Invalid Input

Reverse the adjacent pairs of letters

using System;

namespace myprograms

{

class Program

{

static void Main(string[] args)

{

string input = Console.ReadLine();

Console.WriteLine(UserProgramCode.swapPairs(input));

}

}

}

using System;

namespace myprograms

{

public class UserProgramCode

{

public static String swapPairs(string input)

{

string str = " ";

char temp;

char[] ch = input.ToCharArray();

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

{

if (!char.IsLetter(ch[i]))

{

str = "Invalid Input";

goto l;

}

}

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

{

temp = ch[i];

ch[i] = ch[i + 1];

ch[i + 1] = temp;

i = i + 1;

}

str = new string(ch);

l:return str;

}

}

}

}

}

}

80.Difference between two dates in days  
 Get two date strings as input and write code to find difference between two dates in days.  
    
 Include a class UserProgramCode with a static method getDateDifference which accepts two date strings as input.  
    
 The return type of the output is an integer which returns the diffenece between two dates in days.  
    
 Create a class Program which would get the input and call the static method getDateDifference present in the UserProgramCode.  
    
 Input and Output Format:  
 Input consists of two date strings.  
 Format of date : yyyy-mm-dd.  
    
 Output is an integer.   
 Refer sample output for formatting specifications.  
    
 Sample Input 1:  
 2012-03-12   
 2012-03-14  
 Sample Output 1:  
 2  
 Sample Input 2:  
 2012-04-25   
 2012-04-28  
 Sample Output 2:  
 3

DIFFERENCE B/W TWO DATES

class Program

{

static void Main(string[] args)

{

string s,s1;

s = Console.ReadLine();

s1 = Console.ReadLine();

userprogramcode obj = new userprogramcode();

Console.WriteLine(obj.getDateDifference(s,s1));

}

}

public class userprogramcode

{

public string getDateDifference(string s,string s1)

{

string fm="yyyy-MM-dd";

DateTime dt1,dt;

DateTime.TryParseExact(s,fm,null,System.Globalization.DateTimeStyles.None,out dt);

DateTime.TryParseExact(s1, fm, null, System.Globalization.DateTimeStyles.None, out dt1);

return Convert.ToString((dt1-dt).Days);

}

}

81. Password Encryption  
    
 There is a online shopping site which stores their customer userid and password in their database. However, the password is encrypted before storing them into the DB for maintaining the security. Consider the password and key used for encryption are given as input1 and input2 respectively.Write a function to encrypt based on encryption logic as follows 1) Replace the first letter of every word in input1 with input2. 2) If first letter of any word of input1 matches with input2 then replace the first letter of that word of input1 with the next alphabet and add # symbol next to it. Business Rules : 1)If the password given is empty, then assign Invalid Password to the output1 variable. 2)If the password given consists of digits or special characters,then assign Invalid Input to the output1 variable Create a class named UserProgramCode that has the following static method   
 public static string replaceChar(String input1,String input2)  
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format: Input consists of 2 strings. The first string corresponds to the password and the second string corresponds to the key used.  
 Output consists of a string.  
 Refer sample output and business rule for output formatting specifications.  
    
 Sample Input 1 : Red Green A  
    
 Sample Output 1 : Aed Areen Sample Input 2 : Red Sed Yellow  
 R  
 Sample Output 2 :   
 S#ed Red Rellow

password encryption online shop:

program.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication223

{

class Program

{

static void Main(string[] args)

{

string s1 = Console.ReadLine();

string s2 = Console.ReadLine();

string r = Class1.repl(s1, s2);

if (r == "-1")

{

Console.WriteLine("invalid format");

}

else if (r == "-2")

{

Console.WriteLine("invalid output");

}

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Password

{

class UserProgramCode

{

public static string repl(string s1, string s2)

{

string op = "";

int count = 0;

StringBuilder sb = new StringBuilder();

foreach (char t in s1)

{

count++;

}

if (count == 0)

{

op = "-1";

return op;

}

foreach (char p in s1)

{

if ((char.IsDigit(p)))

//|| (!(char.IsLetterOrDigit(p))))

{

op = "-2";

return op;

}

}

string[] l = s1.Split(' ');

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

{

if (l[i].Substring(0, 1).ToString() != s2)

{

sb.Append(l[i].Replace(l[i].Substring(0, 1), s2));

sb.Append(" ");

}

else if (l[i].Substring(0, 1) == s2)

{

char t = char.Parse(s2);

t = ++t;

string temp = t.ToString();

temp = (char)t + "#";

sb.Append(l[i].Replace(l[i].Substring(0, 1), temp));

sb.Append(" ");

}

}

op = sb.ToString();

return op;

}

}

}

82. Odd Even Sum  
 Write a program to compare the sum of digits at even indexes (say evenSum) and sum of digits at odd indexes (say oddSum) in the given number. Example1: Input1: 23050 (evenSum = 2 + 0 + 0 = 2 oddSum = 3 + 5 = 8) Output = -1 Example2: Input1: 23111 (evenSum = 2 + 1 + 1 = 4 oddSum = 3 + 1 = 4) Output = 1 Business Rule: 1. If both the sums are same then print 1 else print -1.  
    
 Create a class named UserProgramCode that has the following static method  
 public static int sumOfOddEvenPositioned(int input1)  
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 Input consists of an integer.  
 Output is either 1 or -1.  
    
 Sample Input :  
 23050 Sample Output :  
 -1

ODD EVENSUM

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace OddEvenSum

{

class Program

{

static void Main(string[] args)

{

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

int op = UserProgramcode.SumOddEvenIndex(num);

Console.WriteLine(op);

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace OddEvenSum

{

class UserProgramcode

{

public static int SumOddEvenIndex(int num)

{

int rem = 0,sum1=0,sum2=0,k=1;

while (num > 0)

{

if (k % 2 != 0)

{

rem = num % 10;

sum1 = sum1 + rem;

num = num / 10;

k++;

}

else

{

rem = num % 10;

sum2 = sum2 + rem;

num = num / 10;

k++;

}

}

if (sum1 == sum2)

return 1;

else

return -1;

}

}

}

83..Permutations  
    
 Given a String as input, write a program to find all possible permutations of the given input. If the string contains duplicate characters, remove the duplicate characters. Sort the array alphabetically and print the resultant array. The strings in input and output are case sensitive. Business Rule: 1. If string contains any special characters or any digits as input, then print “Invalid Input” .    
 Create a class named UserProgramCode that has the following static method   
 public static List<string> permString(string input1)  
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format:  
 Input consists of a string.  
 Refer business rules and sample output for formatting specifications.  
 Sample Input 1 :  
 cat    
 Sample Output 1 : act atc cat cta tac tca Sample Input 2 :  
 ffin    
 Sample Output 2 : fin fni ifn inf nfi nif Sample Input 3 :  
 ffin%2    
 Sample Output 3 : Invalid Input

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace permutations

{

class UserProgramCode

{

public static List<string> FindPermutations(string set)

{

var output = new List<string>();

if (set.Length == 1)

{

output.Add(set);

}

else

{

var chars = set.ToCharArray();

foreach (var c in chars)

{

var tail = chars.Except(new List<char>() { c });

foreach (var tailPerms in FindPermutations(new string(tail.ToArray())))

{

output.Add(c + tailPerms);

}

}

}

return output;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace permutations

{

class Program

{

static void Main(string[] args)

{

string set = Console.ReadLine();

List<string> output = UserProgramCode.FindPermutations(set);

foreach (var s in output)

{

Console.WriteLine(s);

}

Console.ReadKey();

}

}

}

84.Shortest Length  
 Write a method to get the length of the shortest word in the given string array.    
 Include a class UserProgramCode with a static method shortestWordLength that accepts a string array and returns an integer that corresponds to the length of the shortest word.  
    
 Create a class Program which would get the input and call the static method shortestWordLength present in the UserProgramCode.  
    
 Input and Output Format: First line of the input consists of an integer that corresponds to the number of elements in the string array. The next n lines of the input consists of the elements in the string array. Assume that all the elements in the string array are single words.  
 Output is an integer which corresponds to the length of the shortest word Sample Input 1: 3 cherry hai apple Sample Output 1: 3 Sample Input 2: 4 cherry apple blueberry grapes Sample Output 2: 5

class Program

{

static void Main(string[] args)

{

int i = 0;

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

string[] s = new string[50];

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

s[i] = Console.ReadLine();

s[i] = "\0";

int sl = UserProgramCode.shortestWordLength(s);

Console.WriteLine(sl);

Console.ReadLine();

}

}

class UserProgramCode

{

public static int shortestWordLength(string[] s)

{

int sl =s[0].Length;

for (int i = 1; s[i]!="\0"; i++)

{

if (s[i].Length < sl)

sl = s[i].Length;

}

return sl;

}

}

85..String Encryption  
  Write code to encrypt the given string using following rules and print the encrypted string:  
 Rules:    
 Replace the characters at odd positions by next character in alphabet. Leave the characters at even positions unchanged.  If an odd position charater is 'z' replace it by 'a'. Assume the first character in the string is at position 1. Include a class UserProgramCode with static method encrypt that accepts a string and returns the encrypted string.  
 Create a class Program which would get the input and call the static method encrypt present in the UserProgramCode.  
    
 Input and Output Format :  
 The input is a String . The output is a String which holds the encrypted text. Sample Input 1: curiosity Sample Output 1:  
 dusipsjtz

class Program

{

static void Main(string[] args)

{

string str = Console.ReadLine();

string s1=UserProgramCode.method1(str);

Console.WriteLine(s1);

Console.ReadLine();

}

}

class UserProgramCode

{

public static string method1(string s)

{

int i = 0, a = 0;

StringBuilder sb = new StringBuilder();

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

{

if (i % 2 == 0)

{

if (s[i] == 'z')

sb.Append('a');

else

{

a = Convert.ToInt16(s[i]);

a = a + 1;

sb.Append(Convert.ToChar(a));

}

}

else

sb.Append(s[i]);

}

return sb.ToString();

}

}

86.. Postal Tariff    
 Jack who stays at Delhi sends new year greetings by post to his friends within India. He wants to know the total postal charges he needs to pay for sending the greetings to his friends. There are two types of postal delivery. Normal Post(NP) and Speedy Post (SP). The tariff rates for NP are as follows A. Postal Cost from Delhi to Bhopal(BP) is Rs 100 B. Postal Cost from Delhi to Chennai(CH) is Rs 450 C. Postal Cost from Delhi to Orissa(OS) is Rs 200 For Speedy Post additional 30% of normal Post tariff is charged. The locations and the type of post Jack wants to send are given in the input array where each element of the array is in the format XXYY-where XX represents the location code and YY represents the type of postal delivery done. Write a program to calculate the total cost Jack paid to send the greatings to his friends. Print the output in the following format. Jacks spend Rs ZZZZ to send the greetings where ZZZZ is the total charges calculated. Ignore case sensitivty of input strings. Business rules: 1. If any of the location codes are other than BP,CH or OS,then print "Invalid location Code" . 2. If any of the postal delivery code is other than NP or SP, then print "Invalid Postal Delivery". Create a class named UserProgramCode that has the following static method   
 public static void getPostalTariff(string[] input1)  
    
 Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode. Input and Output Format:    
 The first line of the input consists of an integer, n that corresponds to the number of elements in the string array.  
 The next 'n' lines of input consists of strings that correspond to elements in the string array.  
 Refer business rules and sample output for output format.  
 Always display the total charges to be paid as an int.  
    
 Sample Input 1:  
 3 BPSP CHNP BPNP  
 Sample Output 1: Jack spends Rs 680 to send the greetings Sample Input 2: 3 BPSP CHSP PPNP  
 Sample Output 2:  
 Invalid location Code

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

using System;

using System.Text.RegularExpressions;

namespace code1

{

class Program

{

static void Main(String[] args)

{

int n;

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

String[] input1=new String[n];

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

{

input1[i] = Console.ReadLine();

}

UserMainCode.getPostalTariff(input1);

}

}

}

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

using System;

public class UserMainCode

{

public static void getPostalTariff(string[] input1)

{

int length = input1.Length;

double amount = 0;

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

{

if (input1[i].Substring(2, 2) == "SP")

{

if (input1[i].Substring(0, 2) == "BP")

amount += (100\*1.3);

else if (input1[i].Substring(0, 2) == "CH")

amount += (450 \* 1.3);

else if (input1[i].Substring(0, 2) == "OS")

amount += (200 \* 1.3);

else

{ Console.WriteLine("Invalid location Code"); return; }

}

else if (input1[i].Substring(2, 2) == "NP")

{

if (input1[i].Substring(0, 2) == "BP")

amount += (100);

else if (input1[i].Substring(0, 2) == "CH")

amount += (450);

else if (input1[i].Substring(0, 2) == "OS")

amount += (200);

else

{Console.WriteLine("Invalid location Code");return;}

}

else

{ Console.WriteLine("Invalid Postal Delivery"); return;}

}

Console.WriteLine("Jack spends Rs "+amount+" to send the greetings");

}

}

87.. String Finder  
 Write a program to read three strings which are Searchstring, Str1 and Str2 as input and to find out if Str2 comes after Str1 in the Searchstring. If yes print “Yes” else print “No”.  
    
 Example:   
 input1 = geniousRajKumarDev   
 input2 = Raj  
 input3 = Dev  
 output = Yes  
    
 Include a class UserProgramCode with a static method stringFinder which accepts 3 strings. The return type (Integer) should return 1 if the Str2 comes after Str1 in the Searchstring, else return 2.  
 Create a Class Program which would be used to read 3 strings, and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of three strings which are Searchstring, Str1 and Str2.  
 Output consists of a String, “Yes” or “No”.  
    
 Refer sample output for formatting specifications.  
    
 Sample Input 1:  
 geniousRajKumarDev   
 Raj  
 Dev  
 Sample Output 1:  
 Yes  
    
 Sample Input 2:  
 geniousRajKumarDev   
 Dev  
 Raj  
 Sample Output 2:  
 No

string finder

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication20

{

public class UserMainCode

{

public static int stringFinder(string str, string str1, string str2)

{

int str1\_len = str1.Length;

int str2\_len = str2.Length;

string temp = "";

int st = 0, count1 = 0, count2 = 0;

while (temp != str1)

{

temp = str.Substring(st, str1\_len);

st++;

}

if (temp == str1)

{

count1++;

}

// Console.WriteLine(temp + "\t" + st);

string sub = str.Substring((st + str1\_len - 1));

temp = "";

st = 0;

while (temp != str2)

{

temp = str.Substring(st, str2\_len);

st++;

}

if (temp == str2)

{

count2++;

}

if (count1 == 1 && count2 == 1)

return 1;

else

return 2;

// Console.WriteLine(temp+"\t"+st);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

namespace ConsoleApplication20

{

class Program

{

static void Main(String[] args)

{

string str, str1, str2,ans;

str = Console.ReadLine();

str1 = Console.ReadLine();

str2 = Console.ReadLine();

ans=Convert.ToString(UserMainCode.stringFinder(str, str1, str2));

if (ans.Equals("1"))

Console.WriteLine("Yes");

else

Console.WriteLine("No");

Console.Read();

}

}

}

88..Add non Common Elements  
 Write a program to read two integer arrays and  to add all the non common elements from the 2 integer arrays. Print the final output.   
    
 Example:  
 input1: [7,9,1,0]  
 input2: [10,6,5]  
 Output1:38  
    
 Business Rules:  
 Only positive numbers should be given to the input Lists.  
 1. If the input1 List consists of negative numbers, return -1.  
 2. If the input2 List consists of negative numbers, return -2.  
 3. If the both the input lists consists of negative numbers, return -3.  
    
    
 Include a class UserProgramCode with a static method sumNonCommonElement which accepts the inputs in the following order (input1, size1, input2, size2). The return type (integer) should return output according to the business rules.  
 Create a Class Program which would be used to accept two lists, and call the static method present in UserProgramCode.  
 Input and Output Format:  
 Input consists of n+m+2 integers, where first two integers corresponds to the size of the two array lists, respectively, followed by the corresponding array elements.  
 Output consists of an Integer(the corresponding output), or a String “Input 1 has negative numbers” if the first array contains negative numbers, “Input 2 has negative numbers” if the second array contains negative numbers, or “Both inputs has negative numbers” if both array has negative numbers.  
 Refer sample output for formatting specifications.  
    
 Sample Input 1:  
 4  
 3  
 6  
 9  
 2  
 1  
 10  
 7  
 5  
 Sample Output 1:  
 40  
    
 Sample Input 2:  
 4  
 3  
 -6  
 9  
 2  
 1  
 10  
 7  
 5  
 Sample Output 2:  
 Input 1 has negative numbers  
    
 Sample Input 3:  
 4  
 3  
 6  
 9  
 2  
 1  
 10  
 -7  
 5  
 Sample Output 3:  
 Input 2 has negative numbers  
    
 Sample Input 3:  
 4  
 3  
 6  
 9  
 -2  
 1  
 10  
 -7  
 5  
 Sample Output 3:  
 Both inputs has negative numbers

.Add non Common Elements

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication9

{

public class UserProgramCode

{

public static int sumNonCommonElement(int[] ar1,int n,int[] ar2,int m)

{

int a=0,b=0,c=0;

int sum=0;

int [] temp=new int[m+n];

//List<int> li=new List<int>;

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

if (ar1[i] < 0)

a = 1;

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

if (ar2[j] < 0)

if (a == 0 && b == 0)

{

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

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

if (ar1[i] == ar2[j])

{

ar1[i] = 0;

ar2[j] = 0;

}

return ar1.Sum()+ar2.Sum();

}

if (a == 1 && b == 0)

return -1;

else if (b == 1 && a == 0)

return -2;

if (a == 1 && b == 1)

return -3;

return 0;

}

}

class Program

{

static void Main(string[] args)

{

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

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

int[] ar1=new int[n];

int[] ar2=new int[m];

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

ar1[i] = int.Parse(Console.ReadLine());

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

ar2[i] = int.Parse(Console.ReadLine());

int flag = UserProgramCode.sumNonCommonElement(ar1, n, ar2, m);

if(flag==-1)

Console.WriteLine("Input 1 has negative numbers");

else if(flag==-2)

Console.WriteLine("Input 2 has negative numbers");

else if(flag==-3)

Console.WriteLine("Both inputs has negative numbers");

else

Console.WriteLine(flag);

}

}

}

89.Interleaved Words  
 Given three input strings input1,input2 and input3, write a program to check whether input3 string is an interleaved word of input1 and input2 strings (i.e. input3 = concatenation of input1 and input2 strings,such that input2 string is concatenated at the end of the input1 string eg. if 'game' is input1 and 'center' is input2,input3 should be equivalent to 'gamecenter' and not 'centergame'). If it’s a interleaved word, then print the following : input3 is a interleaved word of input1 and input2 together Replace input1, input2 and input3 with the corresponding inputs. Ignore case sensitiveness in input and use lowercase to print the output. Business rule: 1) If input1 or input2 strings contains any number, then print -1. 2) If both input1 and input2 strings are same, then print -2. 3) If input1 or input2 contains any special characters, then print -3. Create a class named UserProgramCode that has the following static method   
 public static string checkInterleavedword(string input1,string input2,string input3) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.    
 Input and Output Format: Input consists of 3 strings – input1, input2 and input 3.  
 Refer business rules and sample output for output formatting specifications.  
    
 Sample Input 1 : foreign land foreignland    
 Sample Output 1 :   
 foreignland is a interleaved word of foreign and land together Sample Input 2 : string1 set string1set    
 Sample Output 2 :  
 -1  

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Interleaved

{

class Program

{

static void Main(string[] args)

{

string str1 = Console.ReadLine().ToLower();

string str2 = Console.ReadLine().ToLower();

string str3 = Console.ReadLine().ToLower();

string op = UserProgramCode.checkInterleavedword(str1, str2, str3);

if (op == "1")

{

Console.WriteLine("{0} is a interleaved word of {1} and {2} together", str3, str1, str2);

}

else

Console.WriteLine(op);

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Interleaved

{

class UserProgramCode

{

public static string checkInterleavedword(string str1, string str2, string str3)

{

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

{

if (char.IsDigit(str1[i]))

{

return "-1";

}

}

for (int i = 0; i < str2.Length; i++)

{

if (char.IsDigit(str2[i]))

{

return "-1";

}

}

if (str1.Equals(str2))

{

return "-2";

}

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

{

if (!char.IsLetter(str1[i]))

{

return "-3";

}

}

for (int i = 0; i < str2.Length; i++)

{

if (!char.IsLetter(str2[i]))

{

return "-3";

}

}

if (str3 == str1 + str2)

{

return "1";

}

return "0";

}

}

}

90.Cattle Graze  
    
 In a village there is a ground with full of grass where the cattle-rearing people take their cattle to maze in the ground. Assume that the cattle is tied to a tree. Write a program to calculate the area of grass that the cattle can maze. The rope length would be the input and area rounded of two decimal places would be the output.  
 Do not use Math.PI for the value of PI. Use 3.14 directly.   
 Include a class UserProgramCode with a static method calculateArea which accepts an integer. The return type is double. The method returns the area rounded to 2 decimal places.  
 Create a Class Program which would be used to accept Input and call the static method present in UserProgramCode. Use random function in Math library.    
 Input and Output Format:  
 Input consists of the integer value n.   
 Output consists of a double.  
 Refer sample output for formatting specifications.  
 Sample Input 1:  
 3  
 Sample Output 1:  
 28.26

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Reflection;

namespace ConsoleApplication2

{

class Program

{

static void Main(string[] args)

{

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

Console.WriteLine(UserProgramCode.calculateArea(n).ToString("#0.00"));

}

}

}

class UserProgramCode

{

public static double calculateArea(int n)

{

double area = 0;

area = Math.Round((3.14\*n\*n),2);

return area;

}

}

91.Form New Word  
 Write a program to read a string and a positive int (say n) as input and to construct a string with first n and last n characters in the given string. Note - the given string length is >= 2n   
 Example:   
 Input1 = California   
 input2 = 3   
 output = Calnia  
    
 Include a class UserProgramCode with a static method formNewWord() that accepts a string and an integer. The method returns a string. Create a class Program which would get the inputs and call the static method formNewWord() present in the UserProgramCode.  
    
 Input and Output Format: Input consists of a string and an integer. Output is a String that corresponds to the newly formed word.  
 Sample Input 1:  
 California 3  
 Sample Output 1:  
 Calnia

.form new word

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Workout41

{

class UserProgramCode

{

public static string formNewWord(string s,int n)

{

string s1,s2;

int l,n1;

l = s.Length;

if (l > n \* 2)

{

n1 = l - n;

s1 = s.Substring(0, n) + s.Substring(n1, n);

return s1;

}

else

{

return "";

}

}

}

class Program

{

static void Main(string[] args)

{

string s;

int n;

UserProgramCode u = new UserProgramCode();

s = Console.ReadLine();

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

s = UserProgramCode.formNewWord(s,n);

Console.WriteLine(s);

}

}

}