Assignment Related to csharp – submitted by Piyush Sharma

1. q.1Write a C# Sharp program that takes three letters as input and display

them in reverse order.

*Test Data*   
Enter letter: b   
Enter letter: a   
Enter letter: t

*Expected Output* :

t a b

Input:

using System;

public class Reverse

{

public static void Main()

{

char letter, letter1, letter2;

Console.Write("Input letter: ");

letter = Convert.ToChar(Console.ReadLine());

Console.Write("Input letter: ");

letter1 = Convert.ToChar(Console.ReadLine());

Console.Write("Input letter: ");

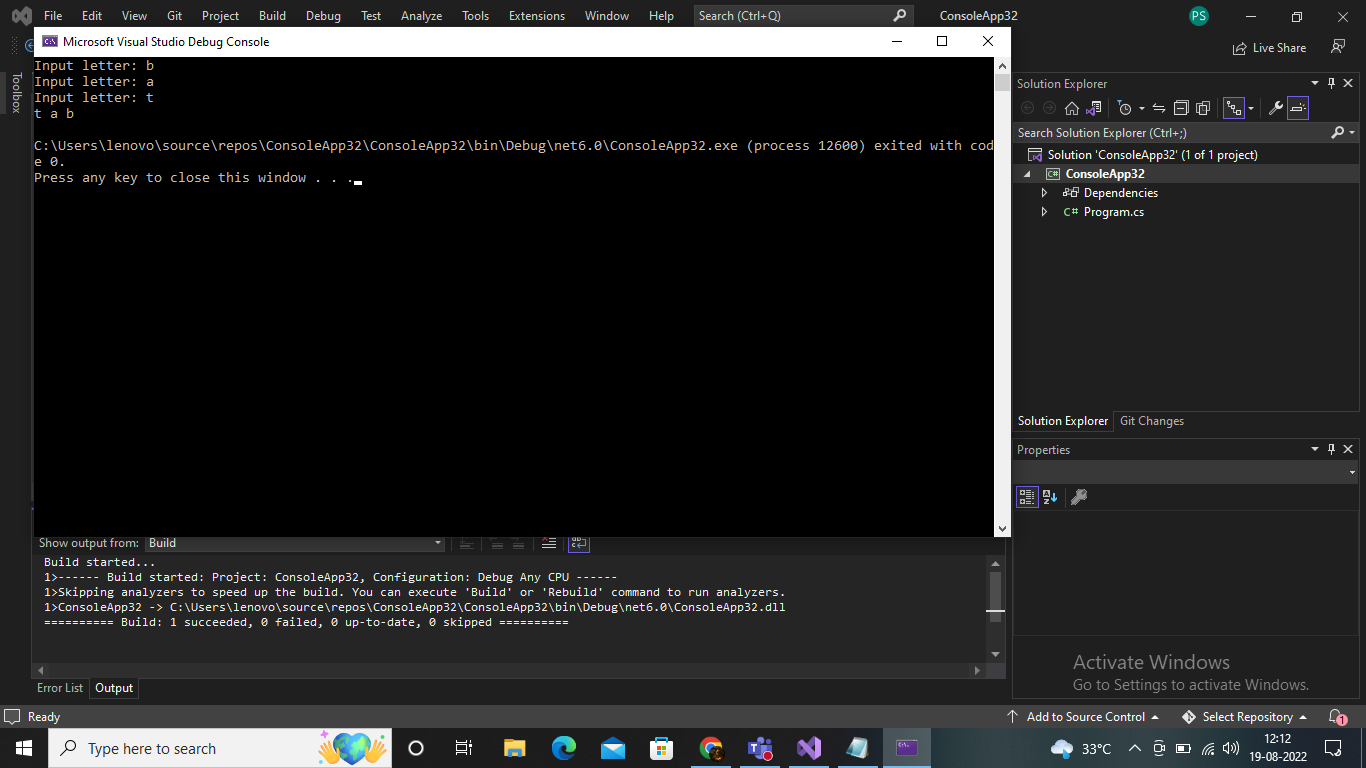
letter2 = Convert.ToChar(Console.ReadLine());

Console.WriteLine("{0} {1} {2}", letter2, letter1, letter);

}

}

Output:



1. Q2; Write a C# Sharp program that takes a number and a width also a number, as input and then displays a triangle of that width, using that number.

*Test Data*   
Enter a number: 6   
Enter the desired width: 6

Expected Output*:*

666666

66666

6666

666

66

6

Input:

using System;

public class Triangle

{

public static void Main()

{

Console.Write("Input a number: ");

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

Console.Write("Input the desired width: ");

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

int height = width;

for (int row = 0; row < height; row++)

{

for (int column = 0; column < width; column++)

{

Console.Write(num);

}

Console.WriteLine();

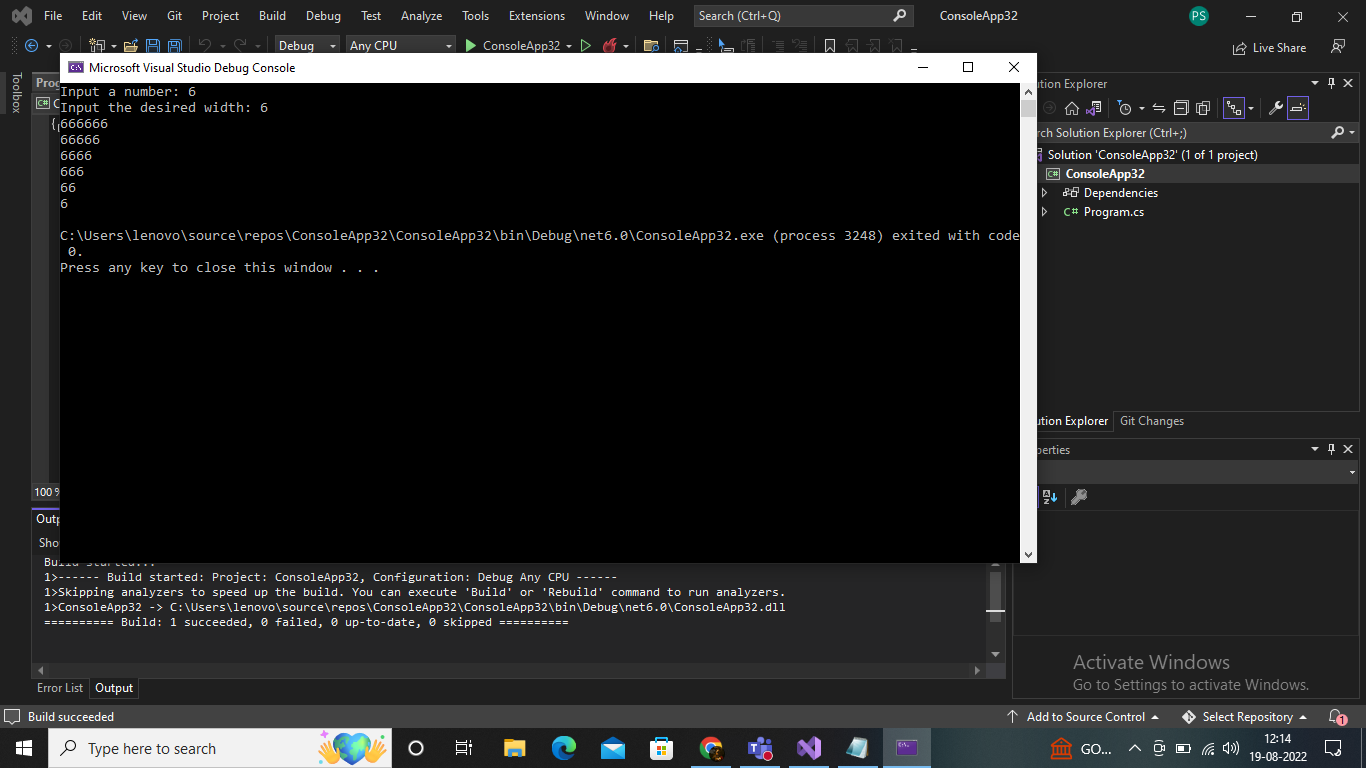
width--;

}

}

}

Output:



3.Write a C# Sharp program that takes userid and password as input (type string). After 3 wrong attempts, user will be rejected

Input: using System;

public class Dictionary

{

public static void Main()

{

string username, password;

int ctr = 0, pd = 0;

Console.Write("User name and password");

Console.Write("Default user name and pw");

do

{

Console.Write("Input a username: ");

username = Console.ReadLine();

Console.Write("Input a password: ");

password = Console.ReadLine();

if (username == "username" && password == "password")

{

pd = 1;

ctr = 3;

}

else

{

pd = 0;

ctr++;

}

}

while ((username != "username" || password != "password")

&& (ctr != 3));

if (pd == 0)

{

Console.Write("Login attemp more than three times failed result");

}

else

if (pd == 1)

{

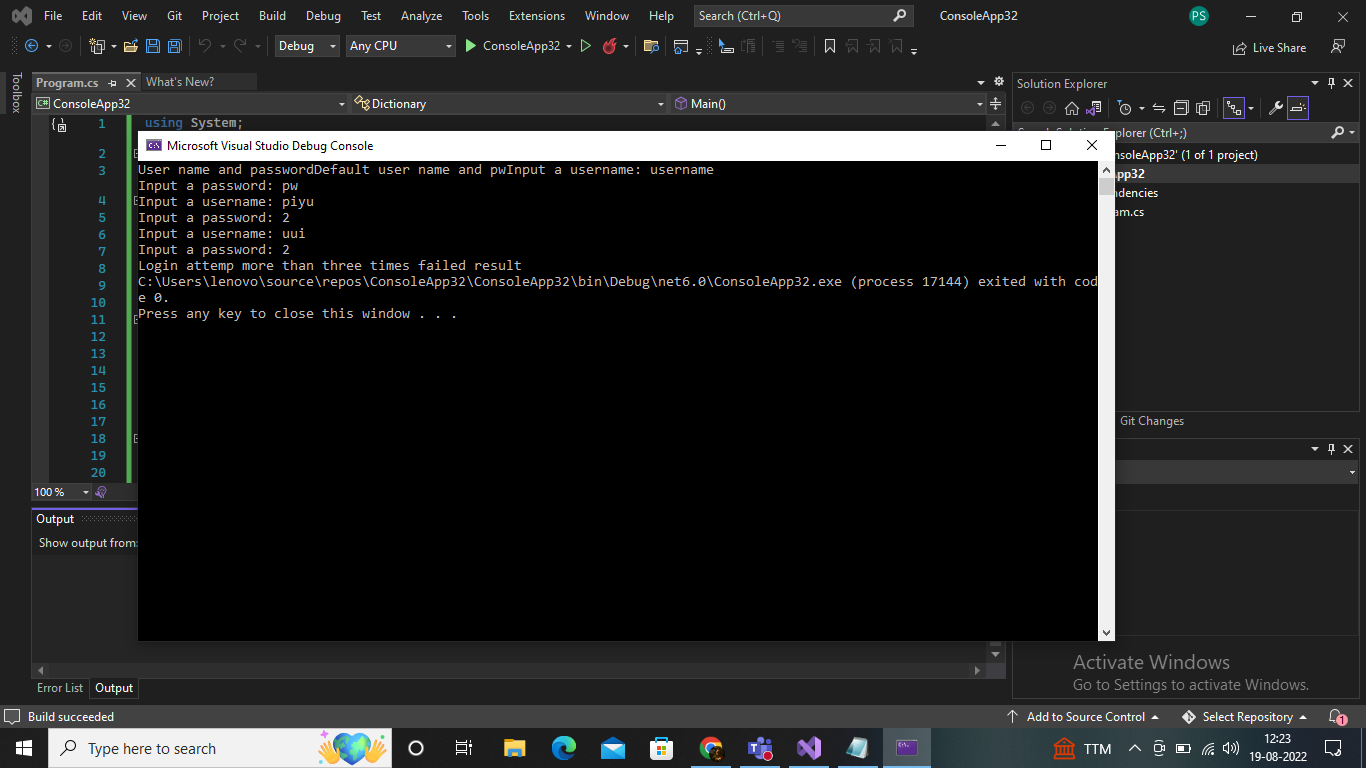
Console.Write("Password entered successfull");

}

}

}

Output:



1. Write a C# Sharp program that takes two numbers as input and perform an operation (+,-,\*,x,/) on them and displays the result of that operation

*Test Data*   
Input first number: 20   
Input operation: -   
Input second number: 12

*Expected Output* :   
20 - 12 = 8

Input:

using System;

public class Calculator

{

public static void Main()

{

int a, b;

char operation;

Console.Write("Enter Integer a: ");

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

Console.Write("Calculator select: ");

operation = Convert.ToChar(Console.ReadLine());

Console.Write("Enter Integer b: ");

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

if (operation == '+')

Console.WriteLine("{0} + {1} = {2}", a, b, a + b);

else if (operation == '-')

Console.WriteLine("{0} - {1} = {2}", a, b, a - b);

else if ((operation == 'x') || (operation == '\*'))

Console.WriteLine("{0} \* {1} = {2}", a, b, a \* b);

else if (operation == '/')

Console.WriteLine("{0} / {1} = {2}", a, b, a / b);

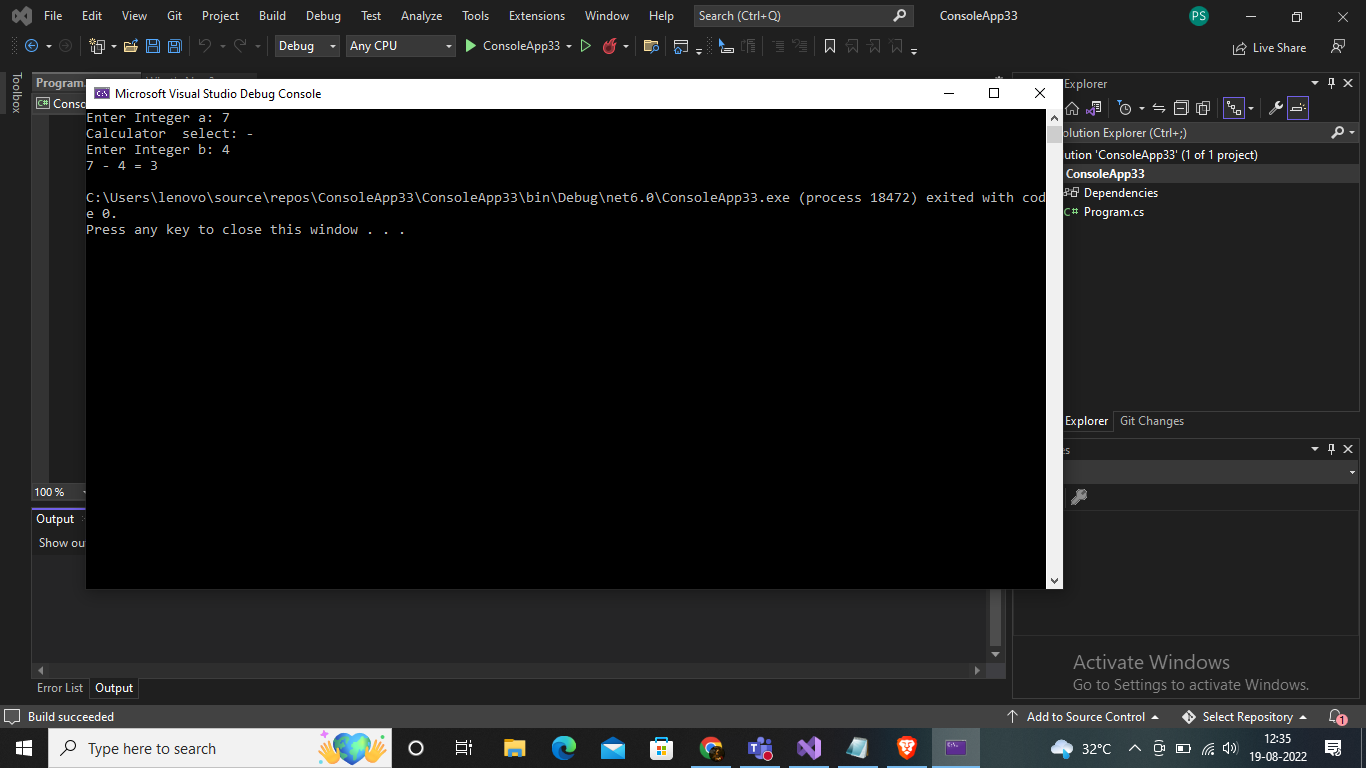
else

Console.WriteLine("Wrong Character");

}

}

Output:



1. Write a C# Sharp program that takes the radius of a circle as input and calculate the perimeter and area of the circle.

*Test Data*   
Input the radius of the circle :   
12

*Expected Output* :   
Perimeter of Circle : 75.36

Input:

using System;

public class Circle

{

public static void Main()

{

double r, per\_cir;

double PI = 3.14;

Console.WriteLine("Input the radius of the circle : ");

r = Convert.ToDouble(Console.ReadLine());

per\_cir = 2 \* PI \* r;

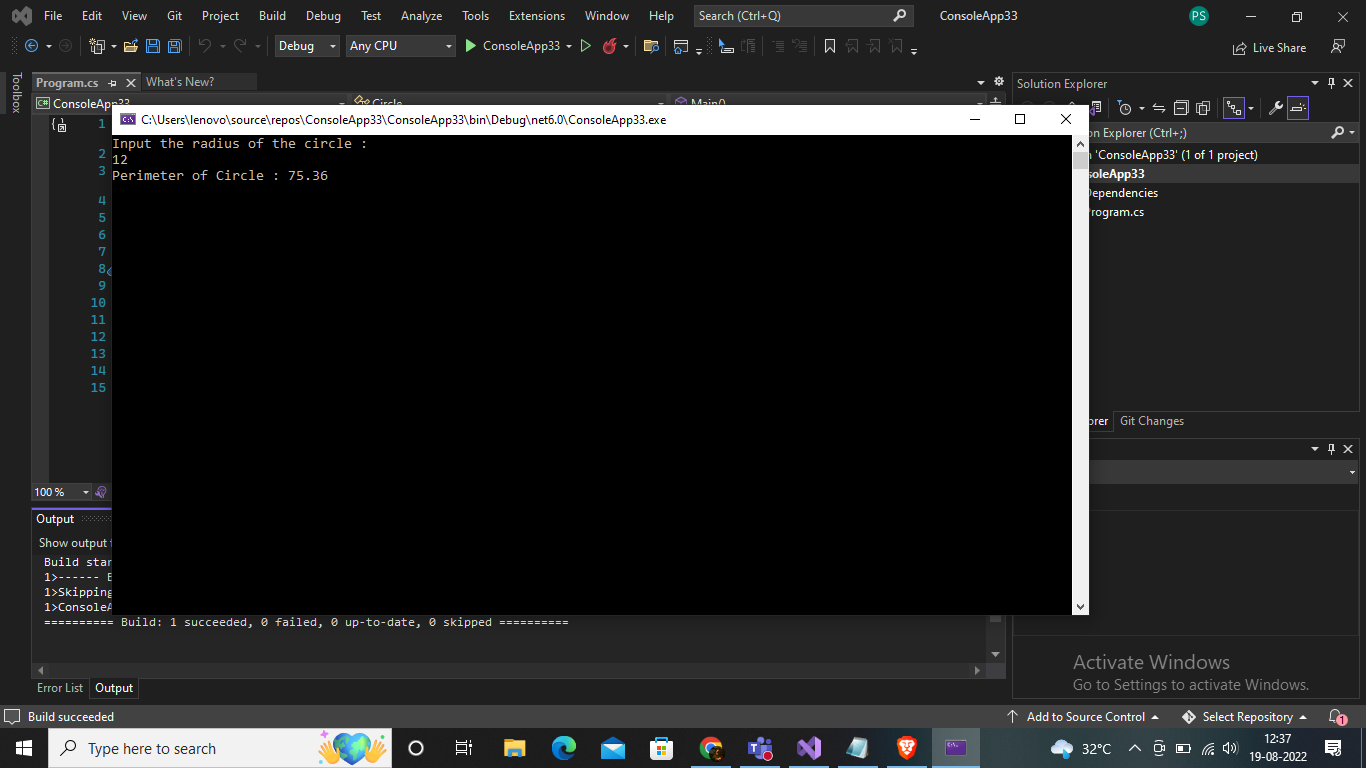
Console.WriteLine("Perimeter of Circle : {0}", per\_cir);

Console.Read();

}

}

Output:



1. Write a C# Sharp program that takes distance and time as input and displays the speed in kilometres per hour and miles per hour.

*Test Data:*   
Input distance(metres): 50000   
Input timeSec(hour): 1   
Input timeSec(minutes): 35   
Input timeSec(seconds): 56

*Expected Output:*   
Your speed in metres/sec is 8.686588   
Your speed in km/h is 31.27172   
Your speed in miles/h is 19.4355

Input:

using System;

public class Exercise7

{

public static void Main()

{

float distance;

float hour, min, sec;

float timeSec;

float mps;

float kph, mph;

Console.Write("Input distance(metres): ");

distance = Convert.ToSingle(Console.ReadLine());

Console.Write("Input timeSec(hour): ");

hour = Convert.ToSingle(Console.ReadLine());

Console.Write("Input timeSec(minutes): ");

min = Convert.ToSingle(Console.ReadLine());

Console.Write("Input timeSec(seconds): ");

sec = Convert.ToSingle(Console.ReadLine());

timeSec = (hour \* 3600) + (min \* 60) + sec;

mps = distance / timeSec;

kph = (distance / 1000.0f) / (timeSec / 3600.0f);

mph = kph / 1.609f;

Console.WriteLine("Your speed in metres/sec is {0}", mps);

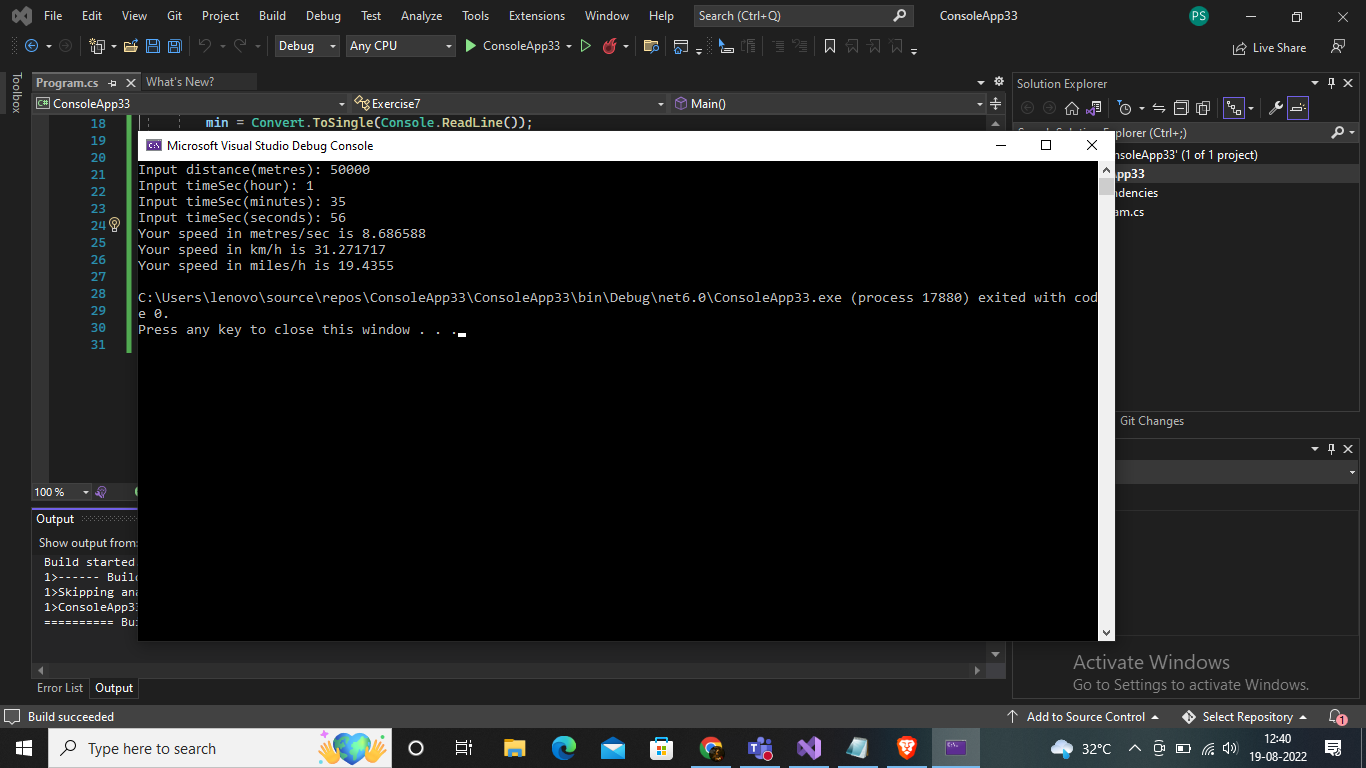
Console.WriteLine("Your speed in km/h is {0}", kph);

Console.WriteLine("Your speed in miles/h is {0}", mph);

}

}

Output



1. Write a C# Sharp program that takes a character as input and check the input (lowercase) is a vowel, a digit, or any other symbol.

*Test Data:*   
Input a symbol: a

*Expected Output:*   
It's a lowercase vowel.

Input:

using System;

public class Vowel

{

public static void Main()

{

char symbol;

Console.Write("Input a symbol: ");

symbol = Convert.ToChar(Console.ReadLine());

if ((symbol == 'a') || (symbol == 'e') || (symbol == 'i') ||

(symbol == 'o') || (symbol == 'u'))

Console.WriteLine("It's a lowercase vowel.");

else if ((symbol >= '0') && (symbol <= '9'))

Console.WriteLine("It's a digit.");

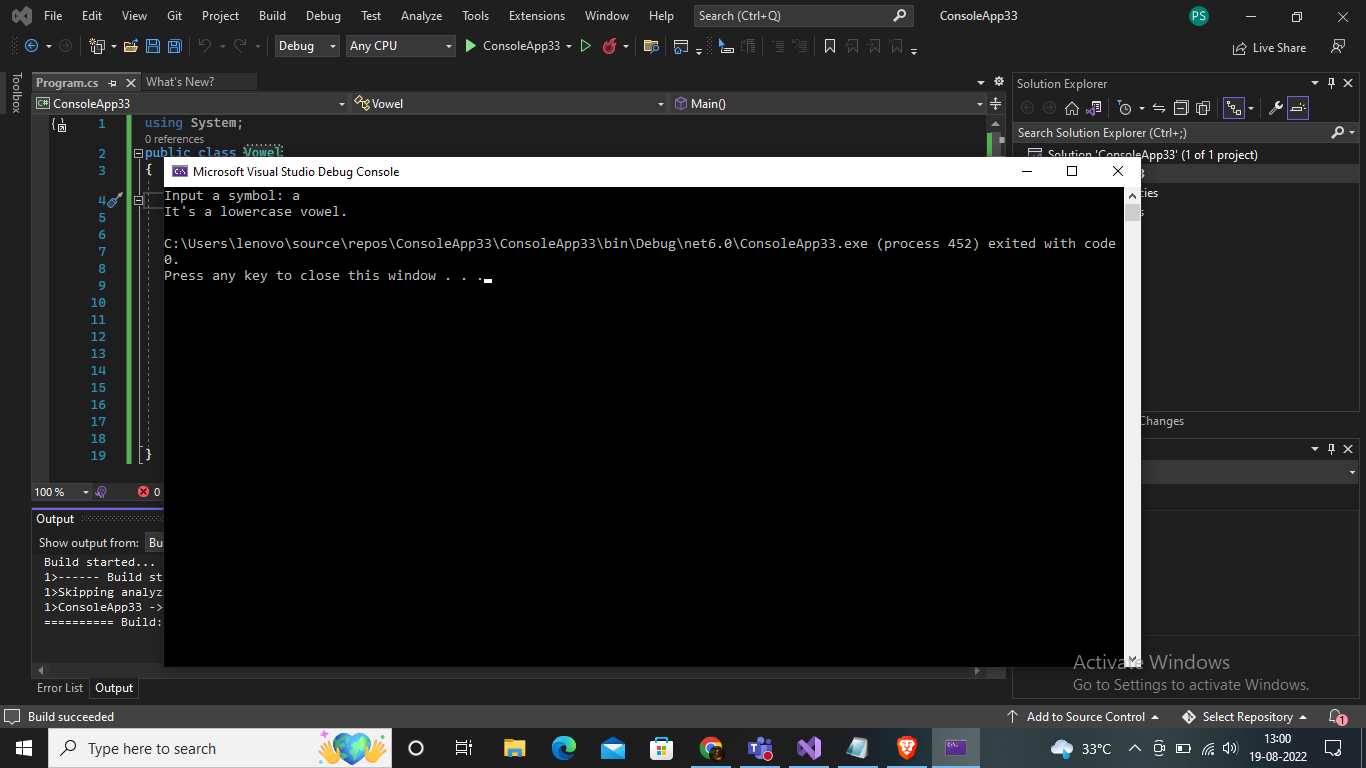
else

Console.Write("It's another symbol.");

}

}

Output;



1. Write a C# Sharp program that takes two numbers as input and returns true or false when both numbers are even or odd.

Input

using System;

public class Evenodd

{

public static void Main()

{

int n1, n2;

bool bothEven;

Console.Write("Enter the first number n1: ");

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

Console.Write("Enter the second number n2: ");

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

bothEven = ((n1 % 2 == 0)

&& (n2 % 2 == 0)) ? true : false;

Console.WriteLine(bothEven ?

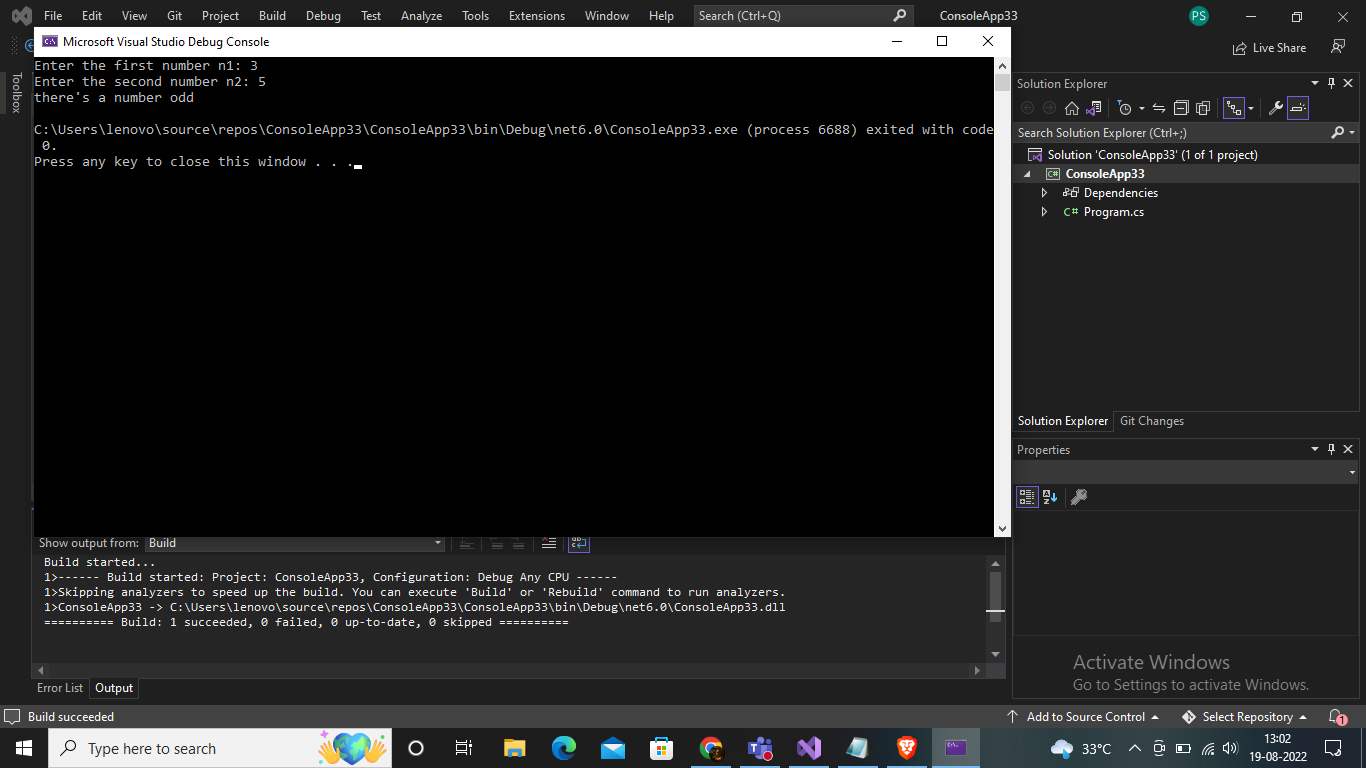
"there're numbers bothEven" :

"there's a number odd");

}

}

Output



1. Write a C# Sharp program that takes a decimal number as input and displays its equivalent in binary form.

*Test Data:*   
Number to convert (or "end")? 25

*Expected Output:*   
Binary: 11001

Input

using System;

class myclass

{

static void Main()

{

int num;

Console.Write("Enter a Number : ");

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

int quot;

string rem = "";

while (num >= 1)

{

quot = num / 2;

rem += (num % 2).ToString();

num = quot;

}

string bin = "";

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

{

bin = bin + rem[i];

}

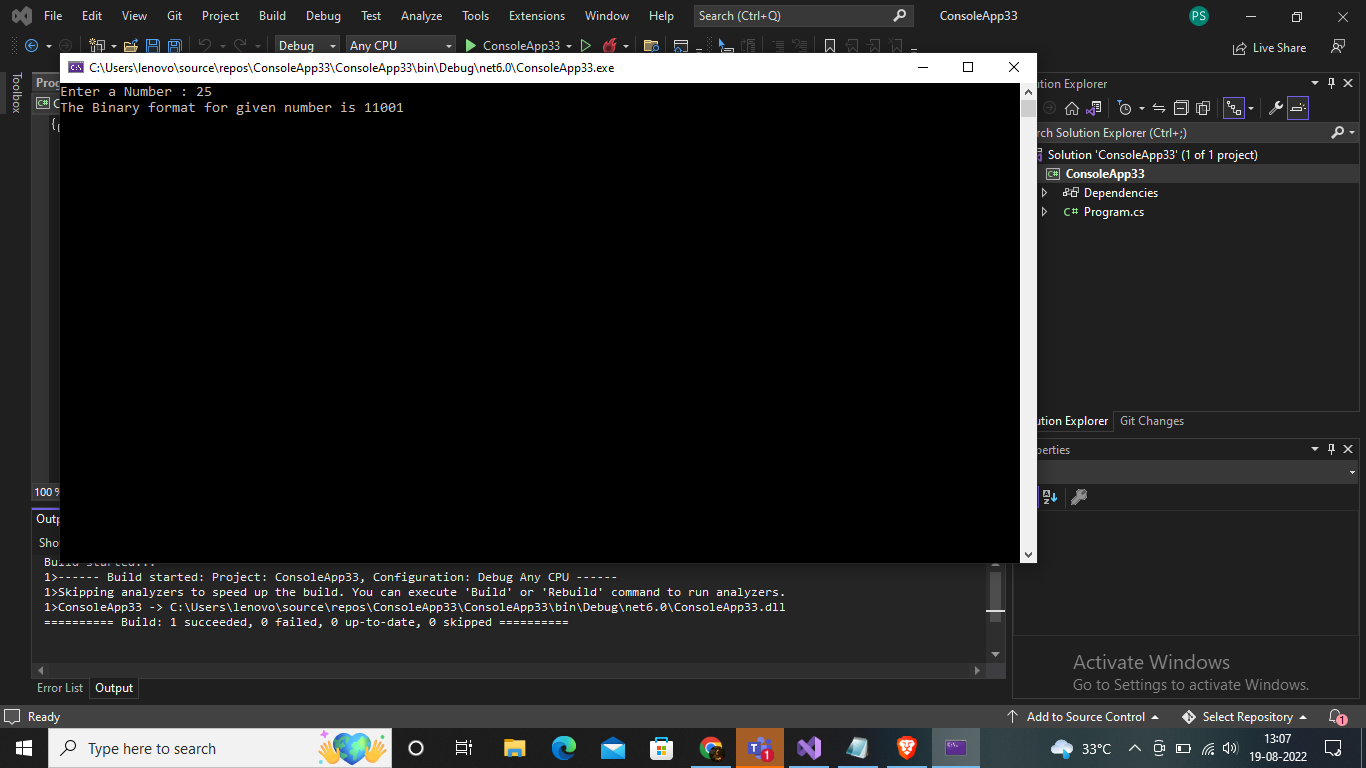
Console.WriteLine("The Binary format for given number is {0}", bin);

Console.Read();

}

}

Output:



1. Write a C# Sharp program to get the absolute difference between n and 51. If n is greater than 51 return triple the absolute difference.

*Sample Input*:   
53   
30   
51   
*Expected Output*:

6

21

0

Input:

using System;

namespace spektra

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine(test(53));

Console.WriteLine(test(30));

Console.WriteLine(test(51));

Console.ReadLine();

}

public static int test(int n)

{

const int x = 51;

if (n > x)

{

return (n - x) \* 3;

}

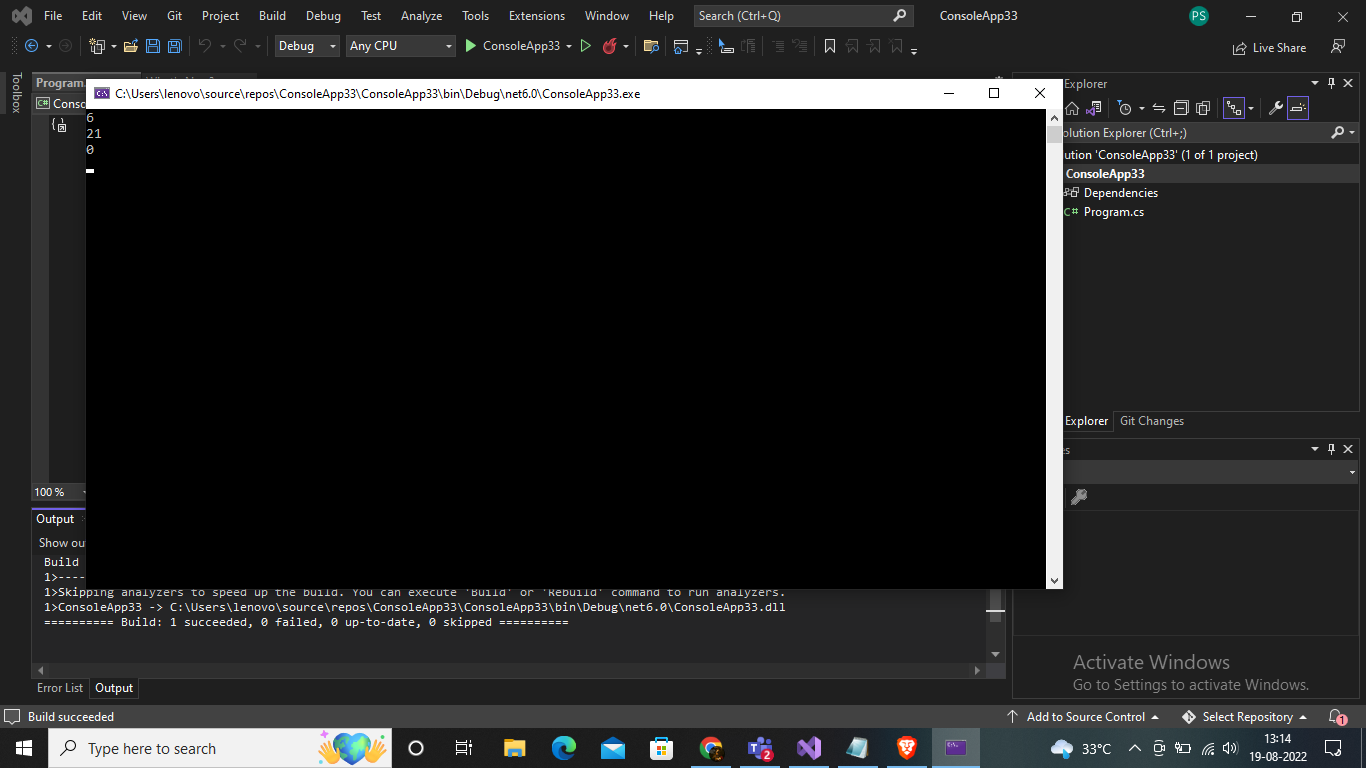
return x - n;

}

}

}

Output:



1. Write a C# Sharp program to remove the character in a given position of a given string. The given position will be in the range 0.. string length -1 inclusive.

*Sample Input*:   
"Python", 1   
"Python", o   
"Python", 4

*Expected Output*:

Pthon

ython

Pythn

Input;

using System;

namespace stringcheck

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine(test("Python", 1));

Console.WriteLine(test("Python", 0));

Console.WriteLine(test("Python", 4));

Console.ReadLine();

}

public static string test(string str, int n)

{

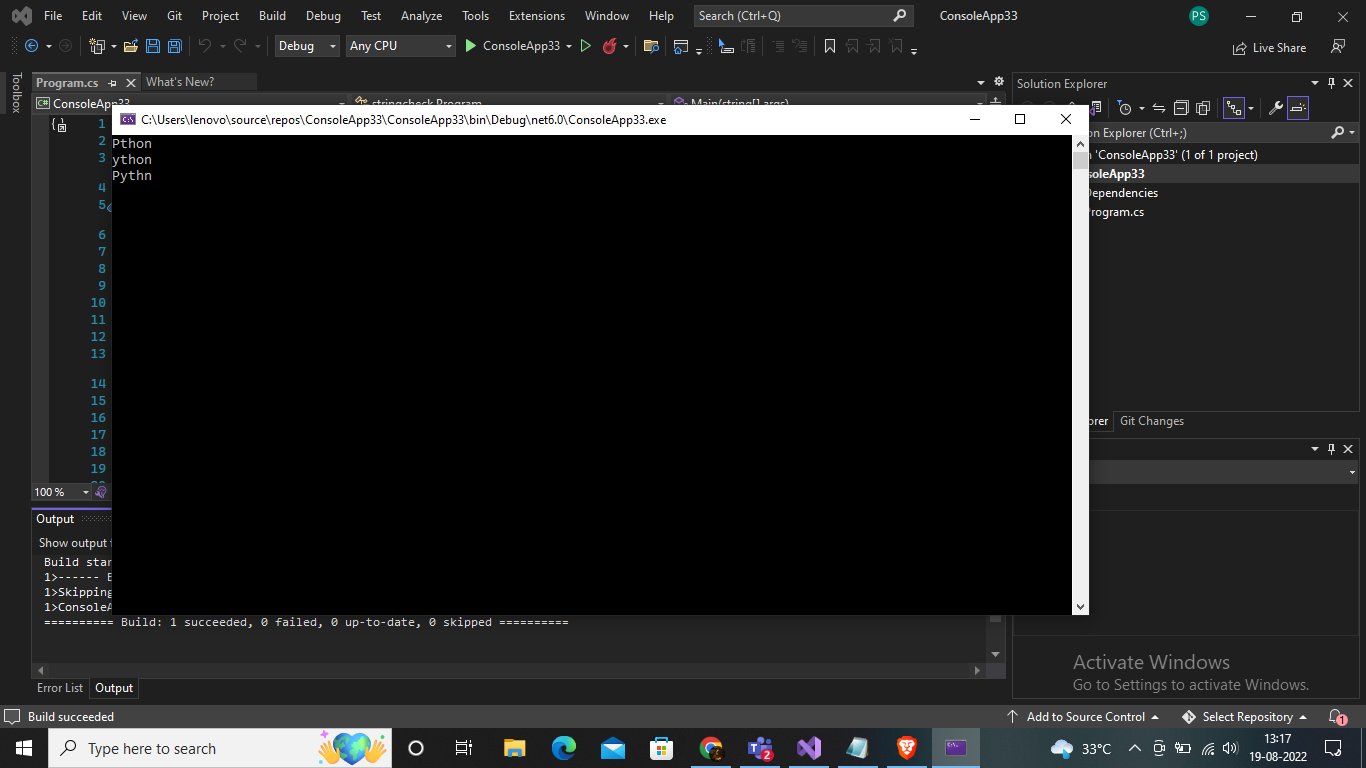
return str.Remove(n, 1);

}

}

}

Output:



1. Write a C# Sharp program to exchange the first and last characters in a given string and return the new string.

*Sample Input*:   
"abcd"   
"a"   
"xy"

*Expected Output*:

dbca

a

yx

Input;

using System;

namespace spektra

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine(test("abcd"));

Console.WriteLine(test("a"));

Console.WriteLine(test("xy"));

Console.ReadLine();

}

public static string test(string str)

{

return str.Length > 1

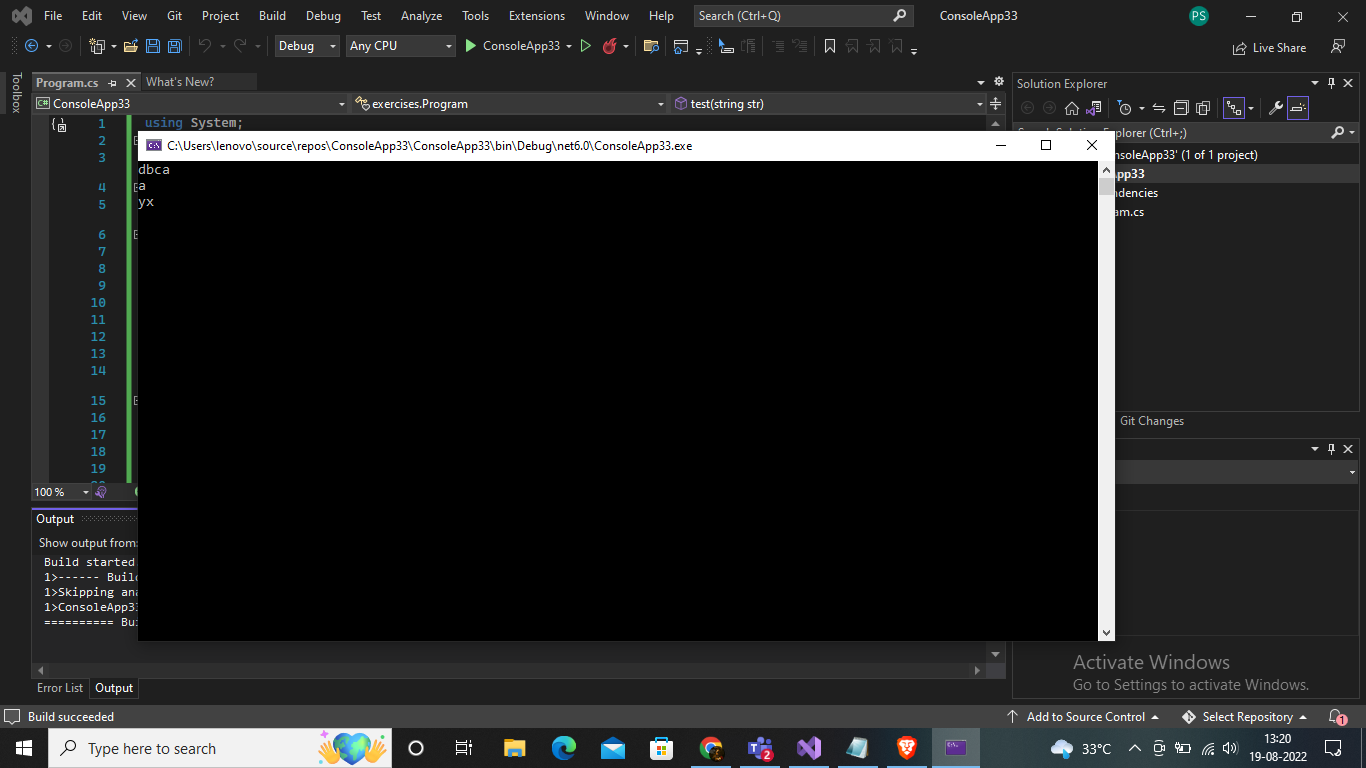
? str.Substring(str.Length - 1) + str.Substring(1, str.Length - 2) + str.Substring(0, 1) : str;

}

}

}

Output;



1. Write a C# Sharp program to create a new array from two give array of integers, each length 3

*Sample Input*:   
{ 10, 20, 30 }, { 40, 50, 60 }

*Expected Output* :

Array 1: 10, 20, 30

Array 2: 40, 50, 60

New array: 10 20 30 40 50 60

Input:

using System;

namespace spektra

{

class Array

{

static void Main(string[] args)

{

int[] item = test(new[] { 10, 20, 30 }, new[] { 40, 50, 60 });

Console.Write("New array: ");

foreach (var i in item)

{

Console.Write(i.ToString() + " ");

}

}

public static int[] test(int[] nums1, int[] nums2)

{

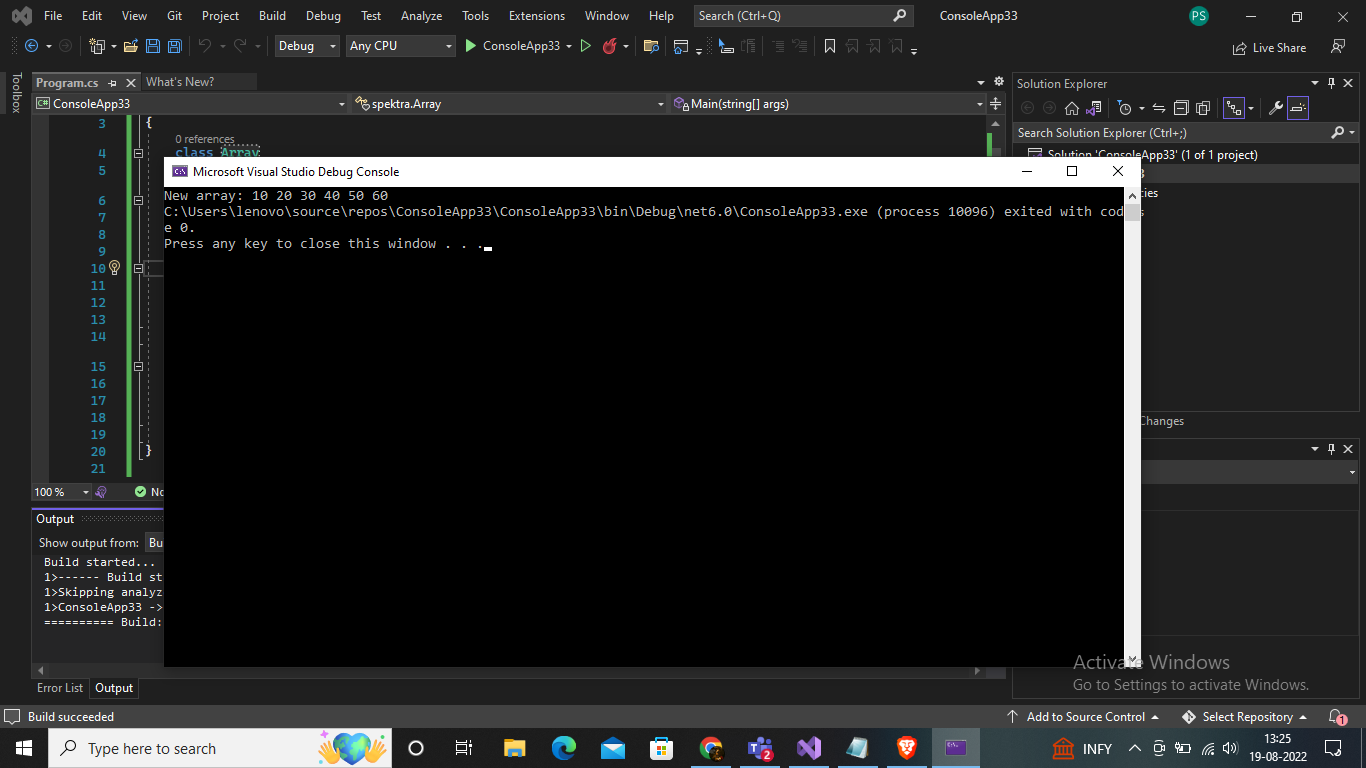
return new int[] { nums1[0], nums1[1], nums1[2], nums2[0], nums2[1], nums2[2] };

}

}

}

Output;



1. Write a C# Sharp program to count the number of strings with given length in given array of strings

*Sample Input:*   
*{"a", "b", "bb", "c", "ccc" }, 1*

*Expected Output:*

*Number of Strings:*

*3*

Input:

namespace spektra

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Number of Strings: ");

Console.WriteLine(test(new[] { "a", "b", "bb", "c", "ccc" }, 1));

}

static int test(string[] arr\_str, int len)

{

int ctr = 0;

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

{

if (arr\_str[i].Length == len) ctr++;

}

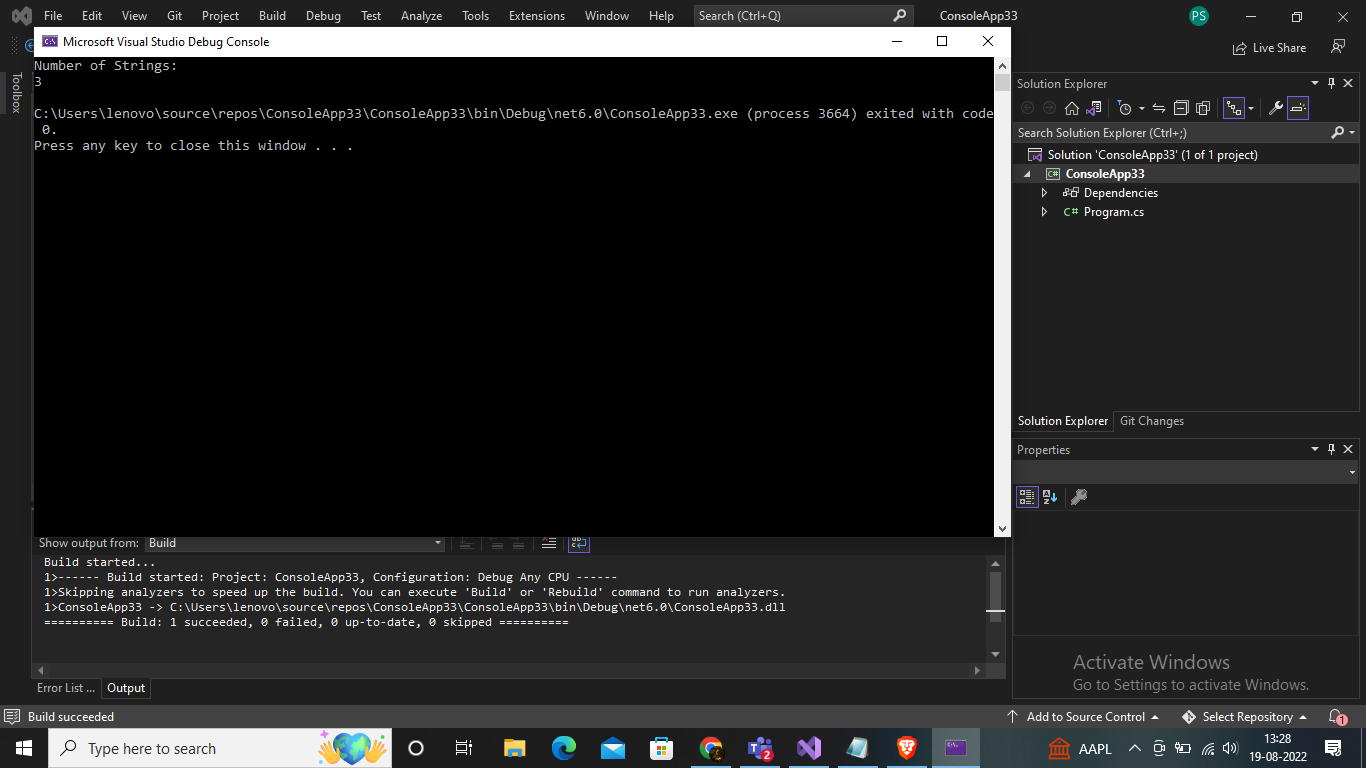
return ctr;

}

}

}

Output;



1. Write a C# Sharp program to calculate the value that results from raising 3 to a power ranging from 0 to 32

Input;

using System;

namespace spektra

{

class Program

{

static void Main(string[] args)

{

int value = 3;

for (int power = 0; power <= 32; power++)

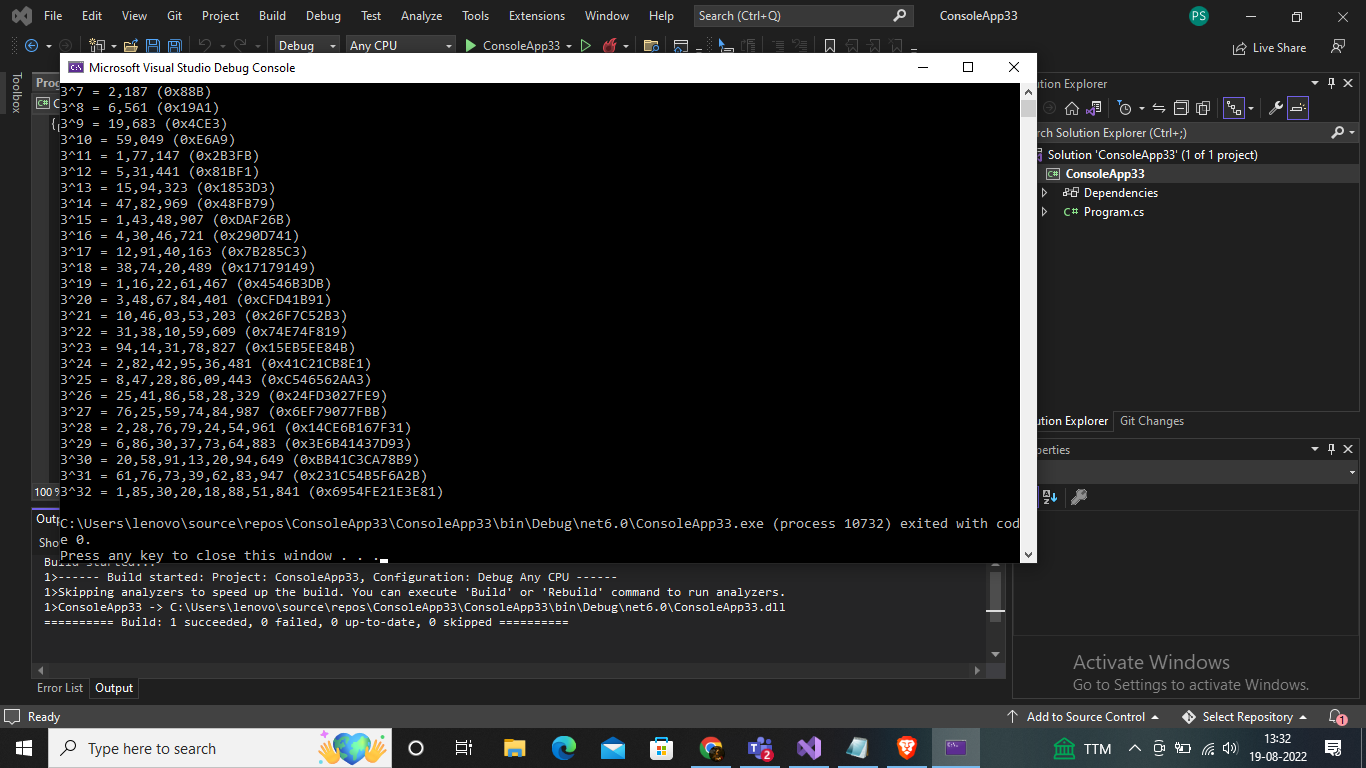
Console.WriteLine($"{value}^{power} = {(long)Math.Pow(value, power):N0} (0x{(long)Math.Pow(value, power):X})");

}

}

}

Output



1. Write a C# Sharp program to convert a given integer value to Roman numerals

*Expected Output:*   
Original integer value: 2365   
Roman numerals of the said integer value:   
MMCCCLXV   
Original integer value: 254   
Roman numerals of the said integer value:   
CCLIV   
Original integer value: 45   
Roman numerals of the said integer value:   
XLV   
Original integer value: 8   
Roman numerals of the said integer value:   
VIII

*Input:*

using System;

using System.Text;

namespace spektra

{

class Program

{

static void Main(string[] args)

{

string s;

s = "MMCCCLXV";

Console.WriteLine("Original integer value: " + s);

Console.WriteLine("Integer value of the said Roman numerals:");

Console.WriteLine(roman\_to\_int(s));

s = "CCLIV";

Console.WriteLine("Original integer value: " + s);

Console.WriteLine("Integer value of the said Roman numerals:");

Console.WriteLine(roman\_to\_int(s));

s = "XLV";

Console.WriteLine("Original integer value: " + s);

Console.WriteLine("Integer value of the said Roman numerals:");

Console.WriteLine(roman\_to\_int(s));

s = "VIII";

Console.WriteLine("Original integer value: " + s);

Console.WriteLine("Integer value of the said Roman numerals:");

Console.WriteLine(roman\_to\_int(s));

}

public static int roman\_to\_int(string str1)

{

var num = 0;

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

{

if (i > 0 && find\_value(str1[i]) > find\_value(str1[i - 1]))

{

num += find\_value(str1[i]) - find\_value(str1[i - 1]) \* 2;

}

else

{

num += find\_value(str1[i]);

}

}

return num;

}

public static int find\_value(char chr)

{

switch (chr)

{

case 'I': return 1;

case 'V': return 5;

case 'X': return 10;

case 'L': return 50;

case 'C': return 100;

case 'D': return 500;

case 'M': return 1000;

default: return 0;

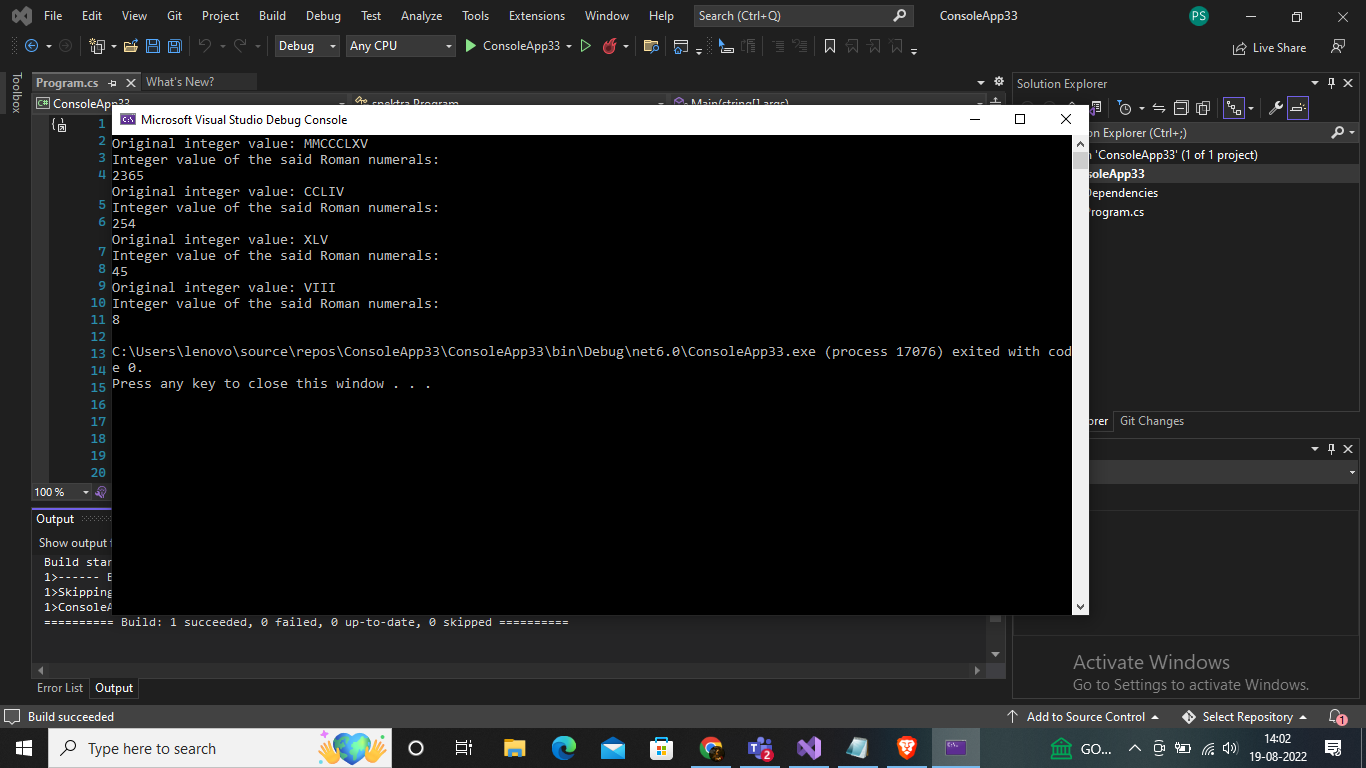
}

}

}

}

Output:



1. Write a program in C# Sharp to find the sum of first n natural numbers using recursion.

Test Data*:*   
How many numbers to sum : 10   
*Expected Output* :   
The sum of first 10 natural numbers is : 55

Input

using System;

public class Recursion

{

static void Main()

{

Console.Write(" Recursion : Display the individual digits of a given number :\n");

Console.Write(" Input any number : ");

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

Console.Write(" The digits in the number {0} are : ", num);

separateDigits(num);

Console.Write("\n\n");

}

static void separateDigits(int n)

{

if (n < 10)

{

Console.Write("{0} ", n);

return;

}

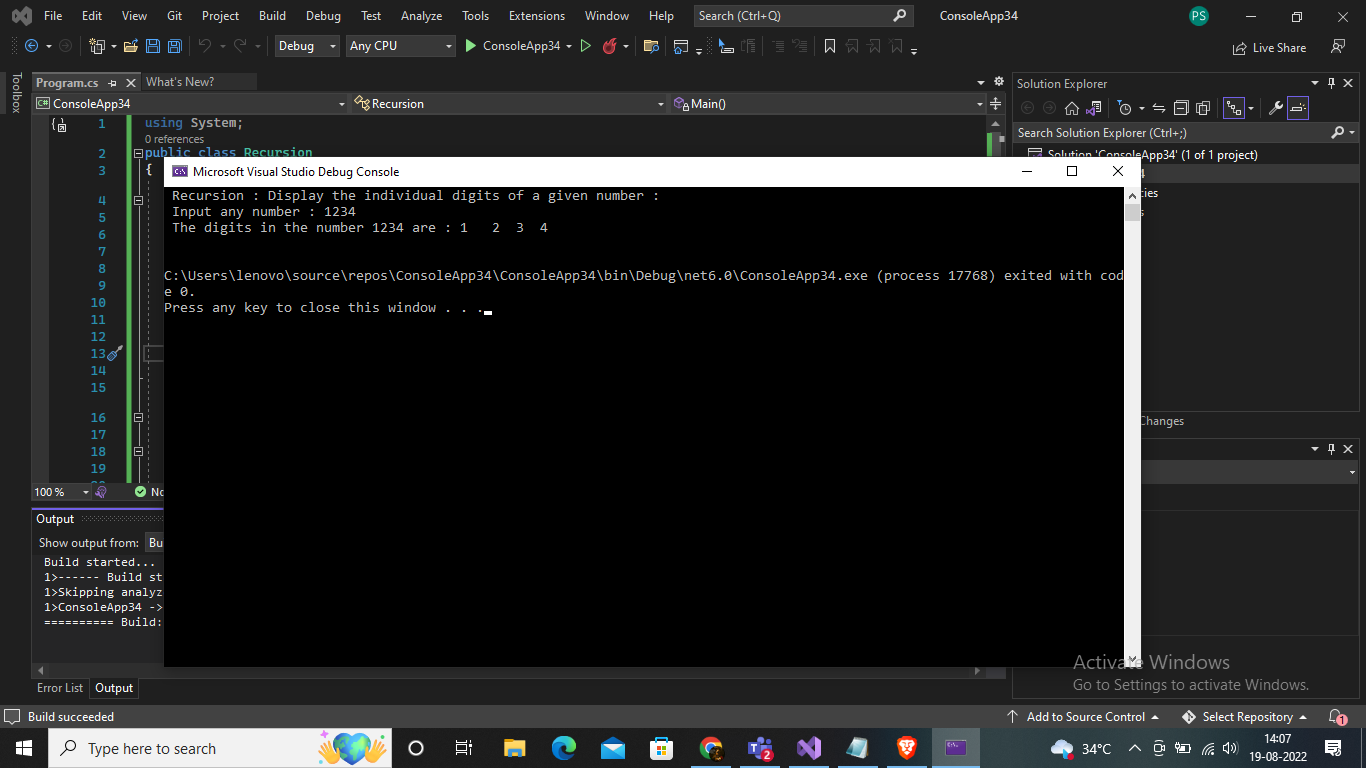
separateDigits(n / 10);

Console.Write(" {0} ", n % 10);

}

}

Output;



1. Write a program in C# Sharp to display the individual digits of a given number using recursion

Test Data*:*   
Input any number : 1234   
*Expected Output* :   
The digits in the number 1234 are : 1 2 3 4

Input;

using System;

using System.Linq;

using System.Collections.Generic;

class LinqExercise

{

static void Main(string[] args)

{

var arr = new[] { 9, 8, 6, 5 };

var sqaure = from int Number in arr

let SqrNo = Number \* Number

select new { Number, SqrNo };

foreach (var a in sqaure)

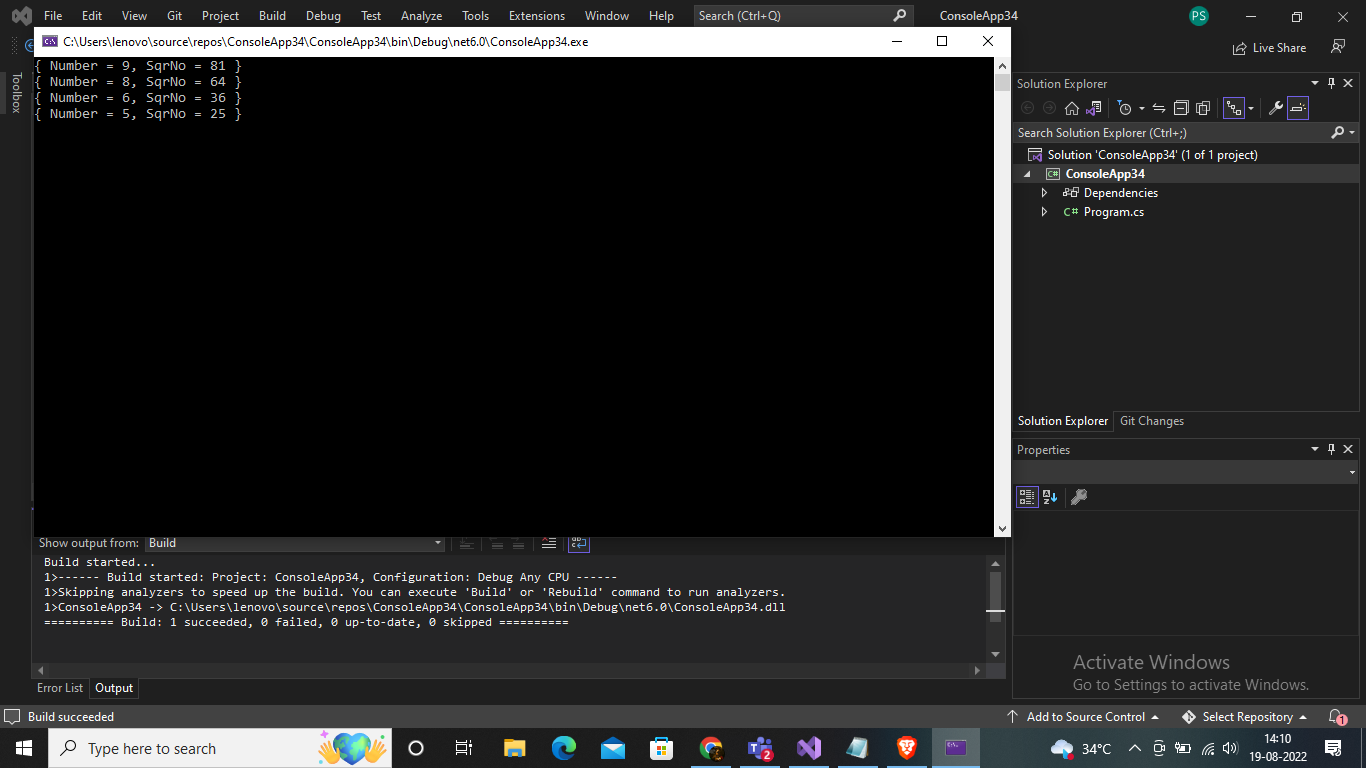
Console.WriteLine(a);

Console.ReadLine();

}

}

Output;



1. Write a program in C# Sharp to find the number of an array and the square of each number using LINQ

*Expected Output* :   
{ Number = 9, SqrNo = 81 }   
{ Number = 8, SqrNo = 64 }   
{ Number = 6, SqrNo = 36 }   
{ Number = 5, SqrNo = 25 }

*Input*

using System;

using System.Linq;

using System.Collections.Generic;

class LinqExercise

{

static void Main(string[] args)

{

string str;

Console.WriteLine("Input the string : ");

str = Console.ReadLine();

var Frequency = from x in str

group x by x into y

select y;

Console.Write("The frequency of the characters are :\n");

foreach (var Arr in Frequency)

{

Console.WriteLine("Character " + Arr.Key + ": " + Arr.Count() + " times");

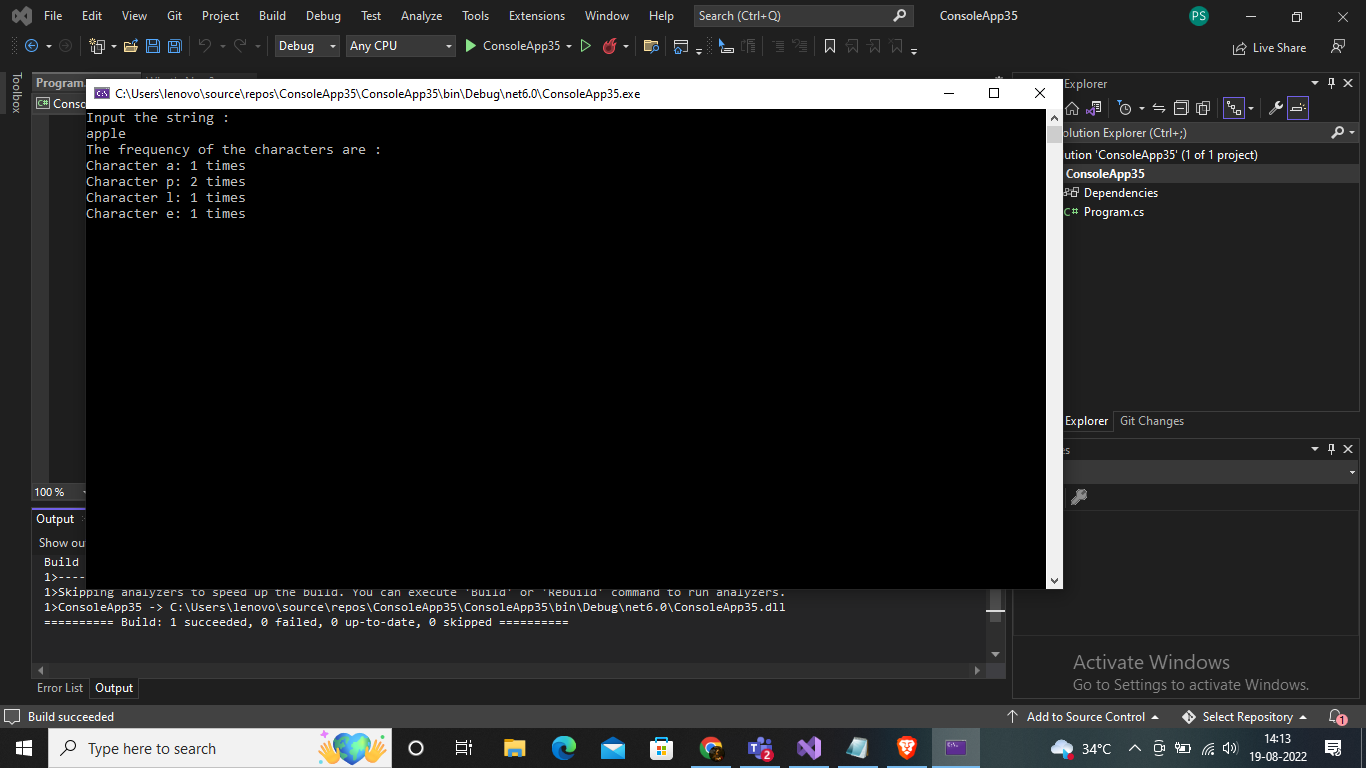
}

Console.ReadLine();

}

}

Output;



21;Write a program in C# Sharp to display the characters and frequency of character from giving string using LINQ

Test Data:   
Input the string: apple

*Expected Output*:   
The frequency of the characters are :   
Character a: 1 times   
Character p: 2 times   
Character l: 1 times   
Character e: 1 times

Input;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

class LinqExercise8

{

static void Main(string[] args)

{

string chst, chen;

char ch;

string[] cities =

{

"ROME","LONDON","NAIROBI","CALIFORNIA","ZURICH","NEW DELHI","AMSTERDAM","ABU DHABI", "PARIS"

};

Console.Write("\nThe cities are : 'ROME','LONDON','NAIROBI','CALIFORNIA','ZURICH','NEW DELHI','AMSTERDAM','ABU DHABI','PARIS' \n");

Console.Write("\nInput starting character for the string : ");

ch = (char)Console.Read();

chst = ch.ToString();

Console.Write("\nInput ending character for the string : ");

ch = (char)Console.Read();

chen = ch.ToString();

var \_result = from x in cities

where x.StartsWith(chst)

where x.EndsWith(chen)

select x;

Console.Write("\n\n");

foreach (var city in \_result)

{

Console.Write("The city starting with {0} and ending with {1} is : {2} \n", chst, chen, city);

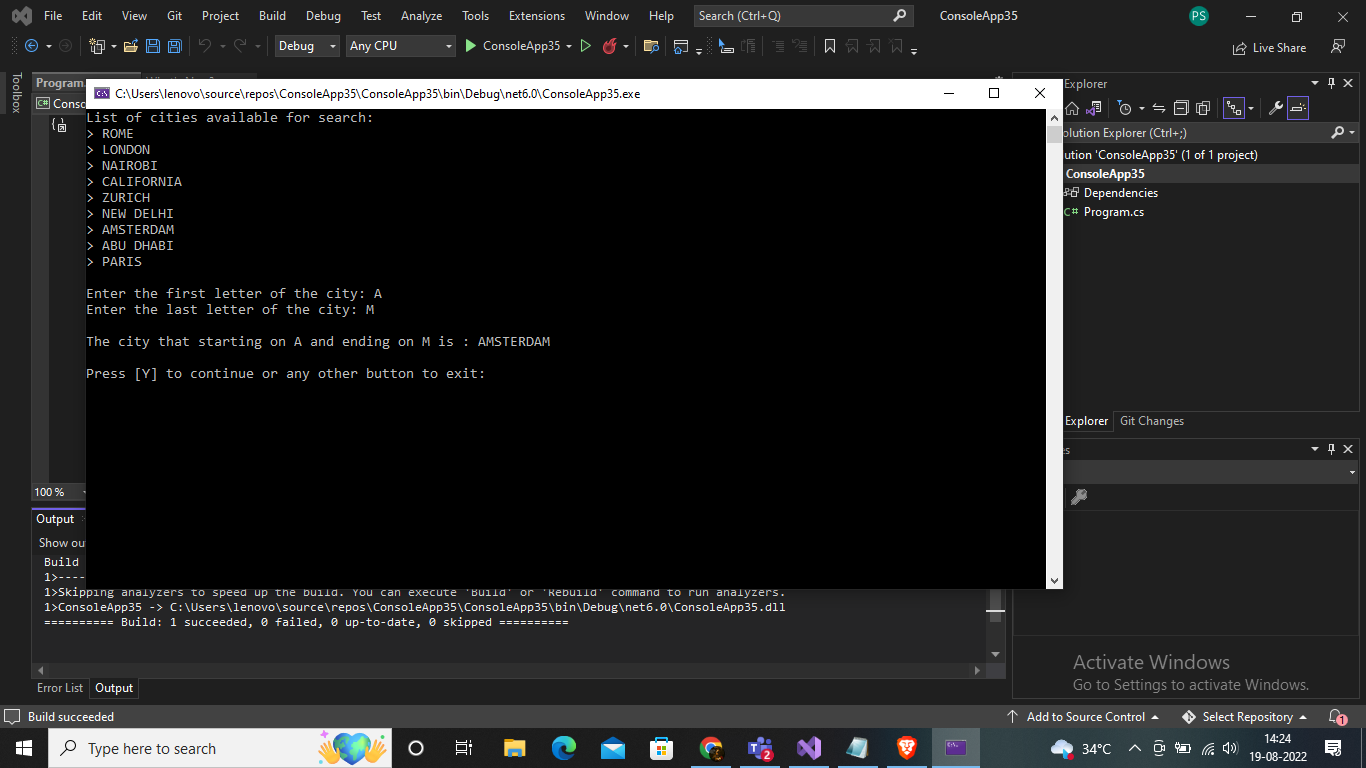
}

Console.ReadLine();

}

}

Output



1. Write a program in C# Sharp to display the top n-th records using LINQ

Test Data :   
The members of the list are :   
5   
7   
13   
24   
6   
9   
8   
7   
How many records you want to display ? : 3   
*Expected Output* :   
The top 3 records from the list are :   
24   
13   
9

Input

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

class Linqnth

{

static void Main(string[] args)

{

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

list.Add(5);

list.Add(7);

list.Add(13);

list.Add(24);

list.Add(6);

list.Add(9);

list.Add(8);

list.Add(7);

Console.WriteLine("\nThe members of the list are : ");

foreach (var lstnum in list)

{

Console.WriteLine(lstnum + " ");

}

Console.Write("How many records you want to display? : ");

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

list.Sort();

list.Reverse();

Console.Write("The top {0} records from the list are: \n", num);

foreach (int topn in list.Take(num))

{

Console.WriteLine(topn);

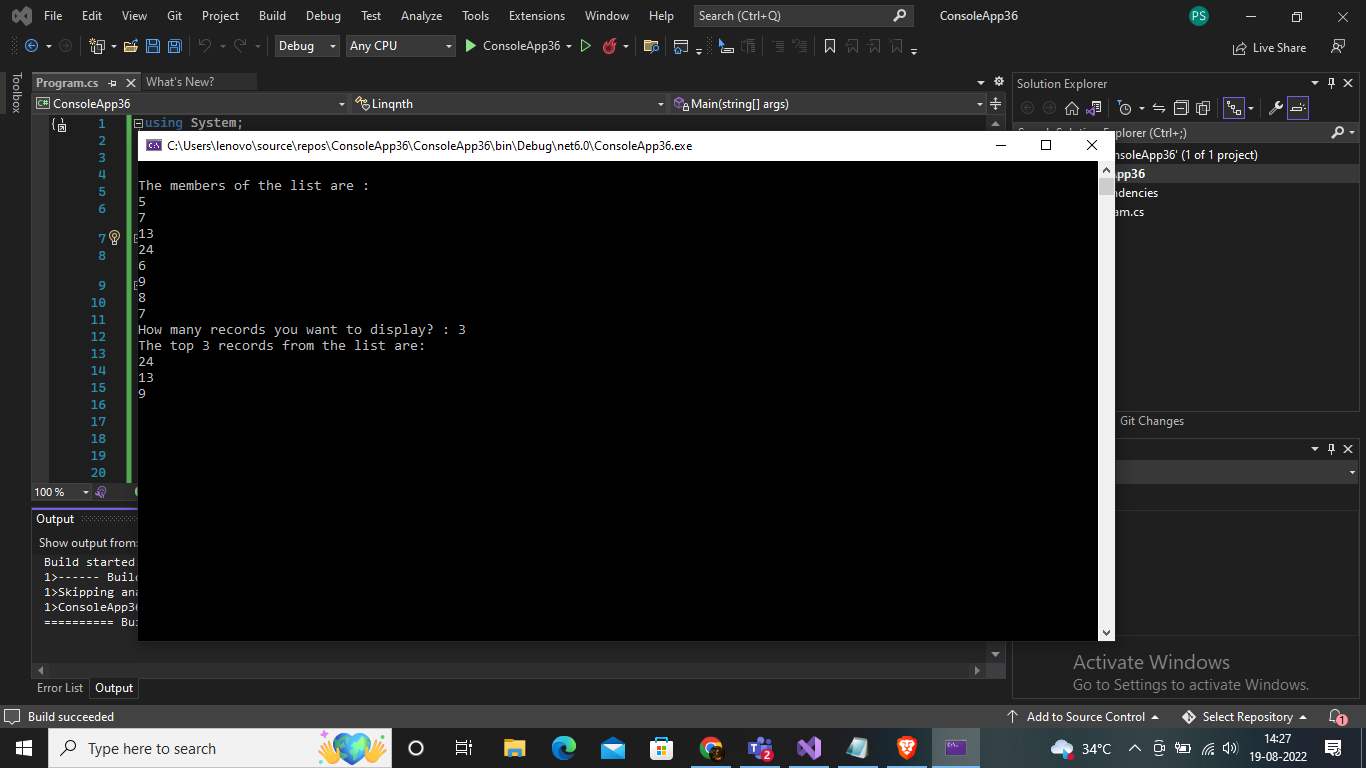
}

Console.ReadLine();

}

}

Output



1. Write a program in C# Sharp to count file extensions and group it using LINQ

Test Data :   
The files are : aaa.frx, bbb.TXT, xyz.dbf,abc.pdf   
aaaa.PDF,xyz.frt, abc.xml, ccc.txt, zzz.txt   
*Expected Output* :   
Here is the group of extension of the files :   
1 File(s) with .frx Extension   
3 File(s) with .txt Extension   
1 File(s) with .dbf Extension   
2 File(s) with .pdf Extension   
1 File(s) with .frt Extension   
1 File(s) with .xml Extension

Input:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

class File

{

public static void Main()

{

string[] arr1 = { "aaa.frx", "bbb.TXT", "xyz.dbf", "abc.pdf", "aaaa.PDF", "xyz.frt", "abc.xml", "ccc.txt", "zzz.txt" };

Console.Write(" Count file extensions and group it : ");

Console.Write("\nThe files are : aaa.frx, bbb.TXT, xyz.dbf,abc.pdf");

Console.Write("\n aaaa.PDF,xyz.frt, abc.xml, ccc.txt, zzz.txt\n");

var fGrp = arr1.Select(file => Path.GetExtension(file).TrimStart('.').ToLower())

.GroupBy(z => z, (fExt, extCtr) => new

{

Extension = fExt,

Count = extCtr.Count()

});

foreach (var m in fGrp)

Console.WriteLine("{0} File(s) with {1} Extension ", m.Count, m.Extension);

Console.ReadLine();

}

}

Output;

