***INDEX***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***S.NO.*** | ***PROGRAM*** | ***PAGE NO.*** | ***DATE*** | ***SIGNATURE*** |
| 1. | Write a C# Program to implement the check that the given number is Armstrong number or not. |  | 20-08-2015 |  |
| 2. | Write a C# Program to implement the find the roots by solving Quadratic Equation (-b +-√b2-4ac) / 2a. |  | 20-08-2015 |  |
| 3. | Write a program to check whether the given input is alphabet, number or special character. |  | 20-08-2015 |  |
| 4. | Write a program to implement Jagged Arrays. |  | 31-08-2015 |  |
| 5. | Write a C# Program to implement Addition, Multiplication, transpose of two matrix (2\*2) and display the sum of diagonal of the resultant matrix. |  | 31-08-2015 |  |
| 6. | Write a program to covert decimal to binary, Octal, Hexadecimal and vice versa. |  | 31-08-2015 |  |
| 7. | Assume that daily temperatures of the last month for four metropolitans are stored in a 2D array. Write a program to calculate city-wise and overall average, max and min. |  | 03-09-2015 |  |
| 8. | Write a program that make use of ref , in , out and param key words . |  | 07-09-2015 |  |
| 9. | Design a class to represent a bank account. Include the following members: Data Members:- Name of the depositor, Account Number, Type of Account, Balance amount in the account and methods : To assign initial values, To deposit an amount, To withdraw an amount after checking balance, To display name and the balance. Write a C# program to demonstrate the working of the various class members |  | 11-9-2015 |  |
| 10. | The student will implement an Hierarchy of classes at least 3 level deep containing a virtual function AND thoroughly understand the override and new keywords (Polymorphism). He must know the order in which the various fields in the hierarchy are initialized and the sequence in which the constructors are called. |  | 14-9-2015 |  |
| 11. | Write a C# Program to implement the use of try, catch for exception handling in a program |  | 18-9-2015 |  |
| 12. | Write a C# program to raise the user defined exception when the given input id not in the range. Range(>18 and <70) Write a C# program to raise the user defined exception when the given input id not in the range. Range(>18 and <70) |  | 18-9-2015 |  |
| 13. | Write a C# program to implement interfaces.(use atleast two interfaces and one class).[ |  | 30-9-2015 |  |
| 14. | Write a program to show the concept of Property and any 10  string functions |  | 1-10-2015 |  |
| 15. | One Complete console application that show the usage of delegates and event mechanism to fire, wire and handle an event |  | 6-10-2015 |  |
| 16. | Write a C# Program to create a statement lambda that returns the factorial of the value it is passed |  | 15-10-2015 |  |
| 17. | Develop a C# Program which is having area of 5 different figures by using two different delegates. |  | 20-10-2015 |  |
| 18. | Write a C# Program that uses order by to retrieve the values from  an object  array of item class.(ascending order of qty) using LINQ. item class must  have item no, item name, qty, price.Create your own class and implement LINQ |  | 3-11-2015 |  |

**CERTIFICATE**

This is to certify that Sunny Khurana of MCA 3rdSemester has successfully completed the practical file on **C Sharp(C#)** for MCA III Practical examination of the GGSIPU in the year 2014. It is further certified that this project is the individual work of the Candidate.

**Mrs. Komal**

Signature:

Date:

ACKNOWLEDGEMENT

I, Sunny Khurana**,** wish to express gratitude to all those who helped and co-operated me and enabled me to complete **C Sharp(C#)** practical file. I express sincere thanks to our teacher, **Mrs. Komal** mam for his valuable suggestion, informative and illuminative guidance.

Through this column, it would be my utmost pleasure to express warm thanks to his encouragement, co-operation and consent

Sunny Khurana

05450404414

Batch 2014 – 2017

**Q.1 Write a C# Program to implement the check that the given number is Armstrong number or not.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication6

{

classProgram

{

staticvoid Main(string[] args)

{

int number, remainder, sum = 0;

Console.Write("enter the Number");

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

for (int i = number; i > 0; i = i / 10)

{

remainder = i % 10;

sum = sum + remainder \* remainder \* remainder;

}

if (sum == number)

{

Console.Write("Entered Number is an Armstrong Number");

}

else

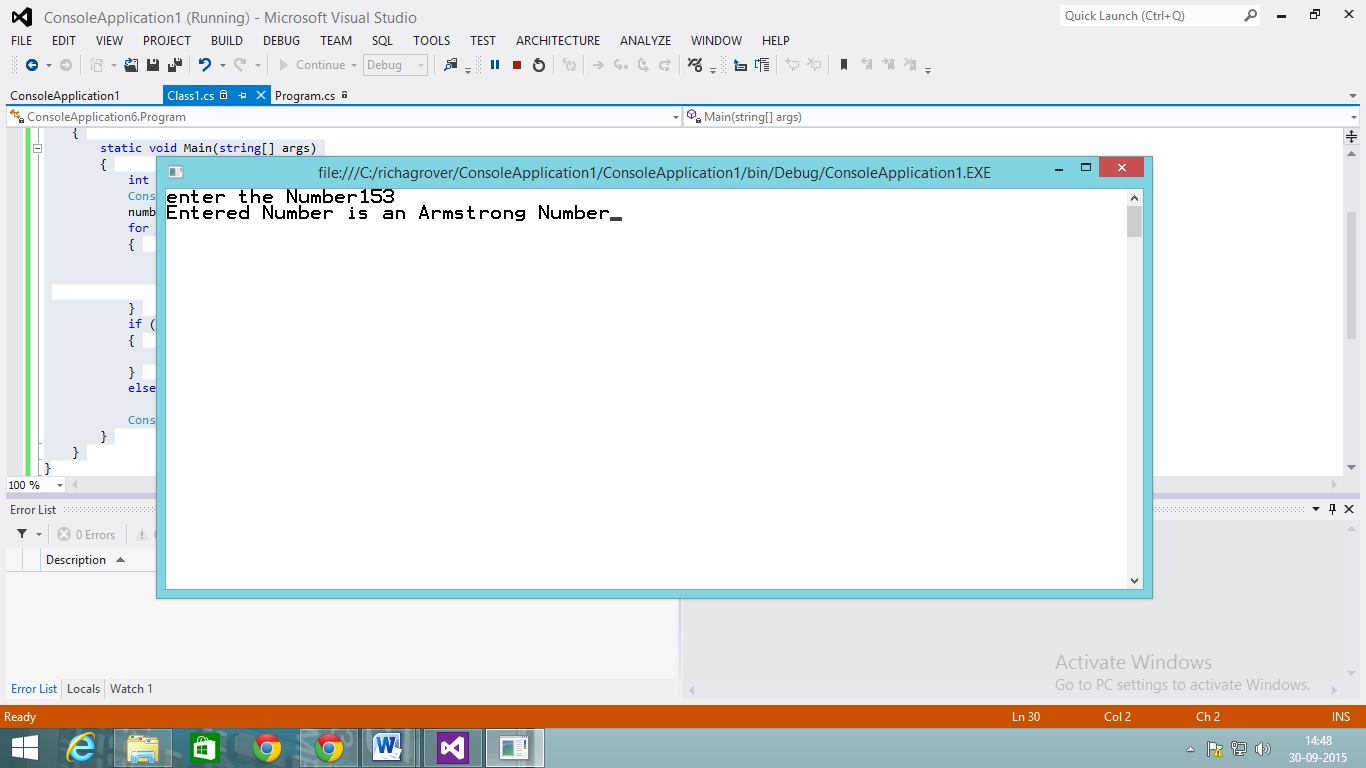
Console.Write("Entered Number is not an Armstrong Number");

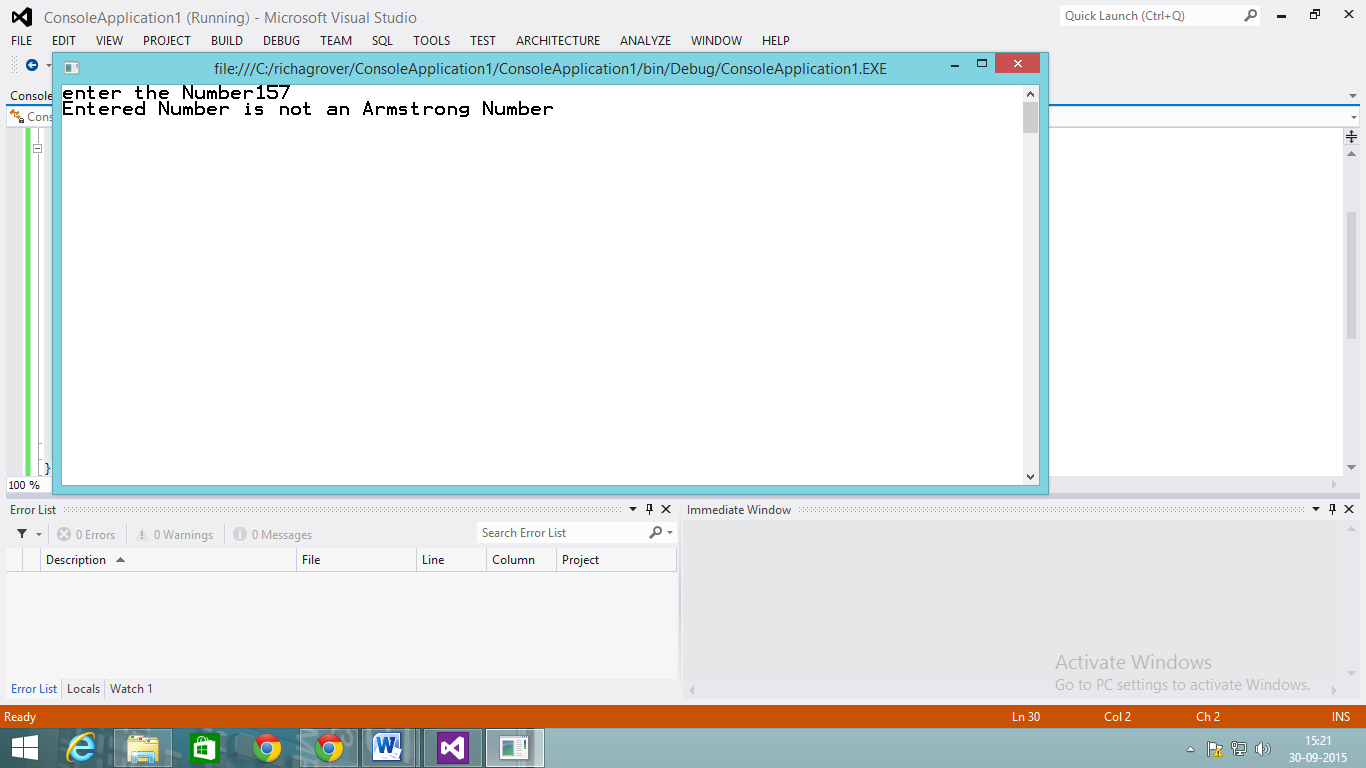
Console.ReadLine();

}

}

}





**Q.2 Write a C# Program to implement the find the roots by solving Quadratic Equation (-b +-√b2-4ac) / 2a.**

using System;

namespace ConsoleApplication1

{

classQuadraticroots

{

double a, b, c;

publicvoid read()

{

Console.WriteLine(" \n To find the roots of a quadratic equation of the form a\*x\*x + b\*x + c = 0");

Console.Write("\n Enter value for a : ");

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

Console.Write("\n Enter value for b : ");

b = double.Parse(Console.ReadLine());

Console.Write("\n Enter value for c : ");

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

}

publicvoid compute()

{

int m;//flag

double r1, r2, d1;

d1 = b \* b - 4 \* a \* c;//to calculate root

if (a == 0)

m = 1;

elseif (d1 > 0)

m = 2;

elseif (d1 == 0)

m = 3;

else

m = 4;

switch (m)

{

case 1: Console.WriteLine("\n Not a Quadratic equation");

Console.ReadLine();

break;

case 2: Console.WriteLine("\n Roots are Real and Distinct");

r1 = (-b + Math.Sqrt(d1)) / (2 \* a);

r2 = (-b - Math.Sqrt(d1)) / (2 \* a);

Console.WriteLine("\n First root is {0:#.##}", r1);

Console.WriteLine("\n Second root is {0:#.##}", r2);

Console.ReadLine();

break;

case 3: Console.WriteLine("\n Roots are Real and Equal");

r1 = r2 = (-b) / (2 \* a);

Console.WriteLine("\n First root is {0:#.##}", r1);

Console.WriteLine("\n Second root is {0:#.##}", r2);

Console.ReadLine();

break;

case 4: Console.WriteLine("\n Roots are Imaginary");

r1 = (-b) / (2 \* a);

r2 = Math.Sqrt(-d1) / (2 \* a);

Console.WriteLine("\n First root is {0:#.##} + i {1:#.##}", r1, r2);

Console.WriteLine("\n Second root is {0:#.##} - i {1:#.##}", r1, r2);

Console.ReadLine();

break;

}

}

}

classRoots

{

publicstaticvoid Main()

{

Quadraticroots qr = newQuadraticroots();

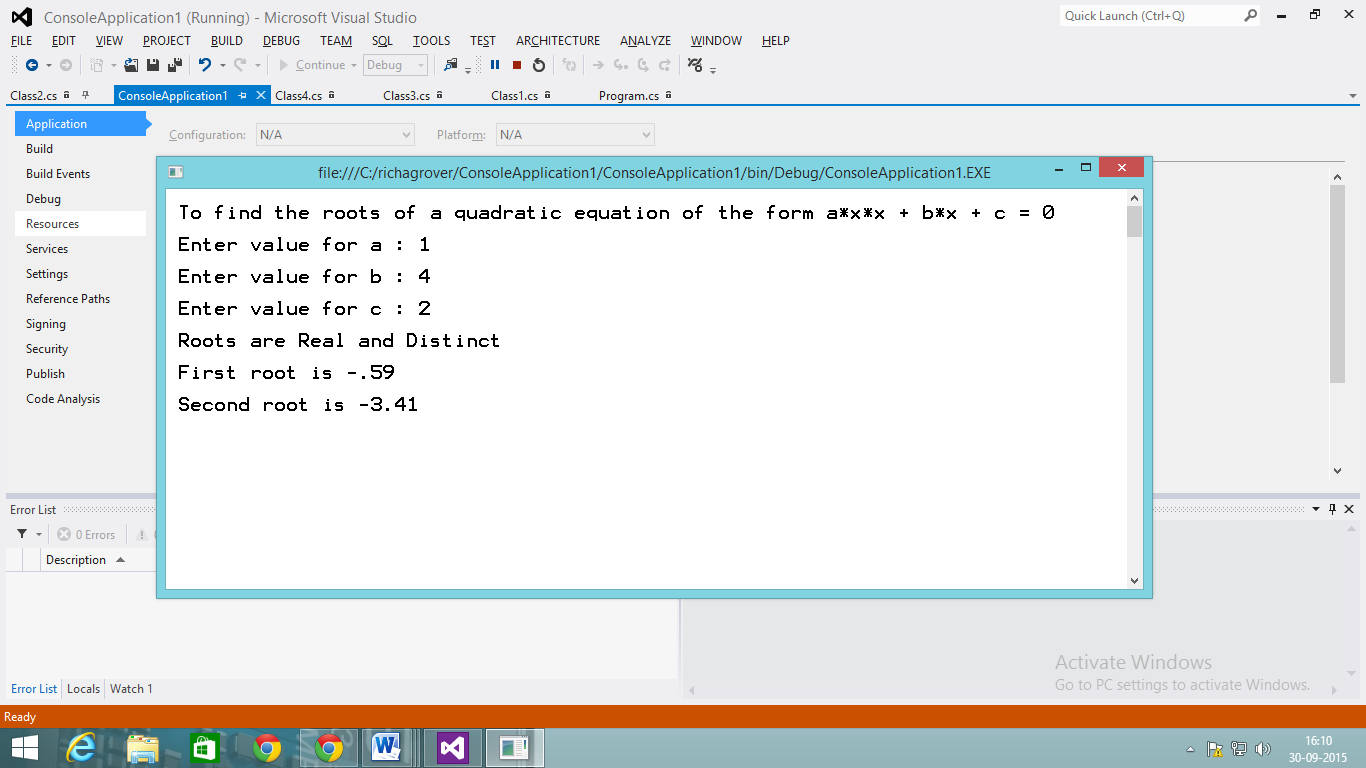
qr.read();

qr.compute();

}

}

}



**Q.3 Write a program to check whether the given input is alphabet, number or special character.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication2

{

classDetect\_Character

{

staticvoid Main()

{

Console.Write("\n Enter any letter: ");

char inputChar = Console.ReadKey().KeyChar;

{

if (Char.IsLetter(inputChar))

{

Console.WriteLine("\n " + inputChar + " is a Letter ");

}

elseif (Char.IsNumber(inputChar))

{

Console.WriteLine("\n " + inputChar + " is a Number ");

}

else

{

Console.WriteLine("\n " + inputChar + " is a special character");

}

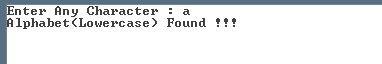
}

Console.ReadKey();

}

}

}

****

**Q.4 Write a program to implement Jagged Arrays.**

using System;

classProgram

{

staticvoid Main()

{

// Declare local jagged array with 3 rows.

int[][] jagged = newint[3][];

// Create a new array in the jagged array, and assign it.

jagged[0] = newint[2];

jagged[0][0] = 1;

jagged[0][1] = 2;

// Set second row, initialized to zero.

jagged[1] = newint[1];

// Set third row, using array initializer.

jagged[2] = newint[3] { 3, 4, 5 };

// Print out all elements in the jagged array.

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

{

int[] innerArray = jagged[i];

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

{

Console.Write(innerArray[a] + " ");

}

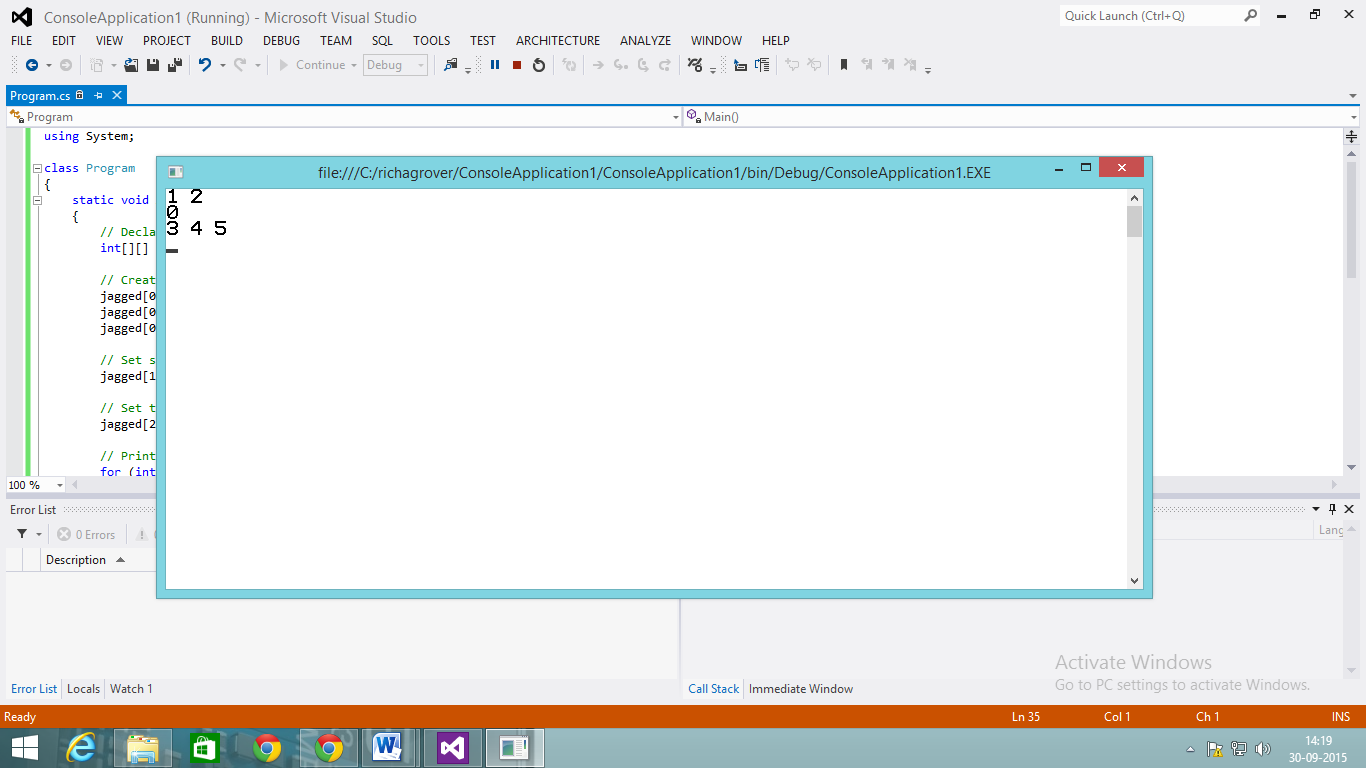
Console.WriteLine();

}

Console.ReadKey();

}

}



**Q.5 Write a C# Program to implement Addition, Multiplication,transpose of two matrix (2\*2) and display the sum of diagonal of the resultant matrix.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Practical\_list

{

class Program5\_Main

{

staticvoid Main(string[] args)

{

char ch;

int [,]arr1=newint[2,2];

int [,]arr2=newint[2,2];

do

{

Console.Clear();

Console.WriteLine("\t\tMENU : ");

Console.WriteLine("\t\t1.Addition ");

Console.WriteLine("\t\t2.Multiplication");

Console.WriteLine("\t\t3.Transpose");

Console.WriteLine("Enter First Matrix(2\*2)");

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

{

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

{

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

}

}

Console.WriteLine("Enter Second Matrix(2\*2)");

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

{

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

{

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

}

}

Console.Write("\nEnter Your Choice : ");

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

switch (choice)

{

case 1:

int[,] sum = newint[2, 2];

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

{

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

{

sum[i, j] = arr1[i, j] + arr2[i, j];

}

}

Console.WriteLine("\nSum of Two Matrix is : ");

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

{

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

{

Console.Write(sum[i, j] + " ");

}

Console.WriteLine();

}

int ad1 = 0, ad2 = 0;

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

{

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

{

if (i == j)

ad1 = ad1 + sum[i, j];

if (i + j == 1)

ad2 = ad2 + sum[i, j];

}

}

Console.WriteLine("\nDiagonal Sum is {0} and {1}", ad1,ad2);

break;

case 2:

int[,] mul = newint[2, 2];

int s = 0;

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

{

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

{

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

{

s = arr1[i, k] \* arr2[k, j] + s;

}

mul[i, j] = s;

s = 0;

}

}

Console.WriteLine("\nMultiplication of Two Matrix is : ");

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

{

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

{

Console.Write(mul[i, j] + " ");

}

Console.WriteLine();

}

int md1 = 0;

int md2 = 0;

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

{

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

{

if (i == j)

md1 = md1 + mul[i, j];

if (i + j == 1)

md2 = md2 + mul[i, j];

}

}

Console.WriteLine("\nDiagonal Sum is {0} and {1}", md1,md2);

break;

case 3:

Console.Write("Which Matrix to be tranpose(1 or 2) ??? ");

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

if (m == 1)

{

int[,] trans = newint[2, 2];

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

{

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

{

trans[j, i] = arr1[i, j];

}

}

Console.WriteLine("\nTranspose matrix is : ");

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

{

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

{

Console.Write(trans[i, j] + " ");

}

Console.WriteLine();

}

int td1 = 0, td2 = 0;

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

{

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

{

if (i == j)

td1 = td1 + trans[i, j];

if (i + j == 1)

td2 = td2 + trans[i, j];

}

}

Console.WriteLine("\nDiagonal Sum is {0} and {1}", td1, td2);

}

elseif (m == 2)

{

int[,] trans = newint[2, 2];

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

{

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

{

trans[j, i] = arr2[i, j];

}

}

Console.WriteLine("\nTranspose matrix is : ");

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

{

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

{

Console.Write(trans[i, j] + " ");

}

Console.WriteLine();

}

int td1 = 0, td2 = 0;

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

{

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

{

if (i == j)

td1 = trans[i, j] + td1;

if (i + j == 1)

td2 = td2 + trans[i, j];

}

}

Console.WriteLine("\nDiagonal Sum is {0} and {1}", td1, td2);

}

else

{

Console.WriteLine("Wrong Choice !!! ");

}

break;

default:

Console.WriteLine("Wrong Choice !!! Try Again ");

break;

}

Console.Write("\nWanna continue ??? : ");

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

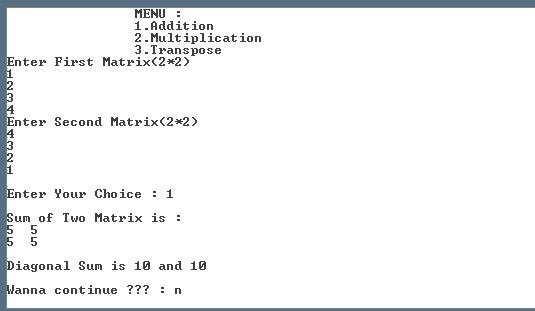
} while ((ch == 'y') || (ch == 'Y'));

}

}

}

**OUTPUT :**



**Q.6 Write a program to covert decimal to binary, Octal, Hexa-decimal and vice versa.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Practical\_list

{

class Program6

{

staticvoid Main(string[] args)

{

int[] arr1 = newint[10];

int[] arr2 = newint[10];

char[] arr3 = newchar[10];

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

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

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

int dec1 = dec, dec2 = dec;

while (dec != 0)

{

arr1[a++] = dec % 2;

dec = dec / 2;

}

Console.Write("\nDecimal to Binary Conversion is : ");

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

{

Console.Write(arr1[i]);

}

Console.Write("\nEquivalent Binary to Decimal Conversion is : ");

int res1 = 0, r = 0;

for (int i1 = 0; i1 < a; i1++)

{

res1 = res1 + ((int)Math.Pow(2, r) \* arr1[i1]);

r++;

}

Console.Write(res1);

while (dec1 != 0)

{

arr2[b++] = dec1 % 8;

dec1 = dec1 / 8;

}

Console.Write("\n\nDecimal to octal Conversion is : ");

for (int j = b - 1; j >= 0; j--)

{

Console.Write(arr2[j]);

}

Console.Write("\nEquivalent octal to decimal Conversion is : ");

int res2 = 0, p = 0;

for (int i2 = 0; i2 < a; i2++)

{

res2 = res2 + ((int)Math.Pow(8, p) \* arr2[i2]);

p++;

}

Console.Write(res2);

while (dec2 != 0)

{

int mod;

mod = dec2 % 16;

if (mod < 10)

mod = mod + 48;

else

mod = mod + 55;

arr3[c++] = (char)mod;

dec2 = dec2 / 16;

}

Console.Write("\n\nDecimal to Hexa Conversion is : ");

for (int k = c - 1; k >= 0; k--)

{

Console.Write(arr3[k]);

}

Console.Write("\nEquivalent Hexa to decimal Conversion is : ");

int res3 = 0, q = 0;

for (int i3 = 0; i3 < c; i3++)

{

if (arr3[i3] <= 57)

{

res3 = res3 + (((int)(arr3[i3] - 48)) \* ((int)Math.Pow(16, q)));

q++;

}

else

{

res3 = res3 + (((int)(arr3[i3] - 55)) \* ((int)Math.Pow(16, q)));

q++;

}

}

Console.Write(res3);

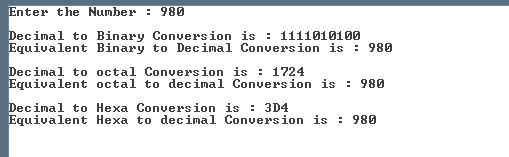
Console.ReadKey();

}

}

}

**OUTPUT :**



**Q.7 Assume that daily temperatures of the last month for four metropolitans are stored in a 2D array. Write a program to calculate city-wise and overall average, max and min.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

classProgram

{

void temp(int[,] arr)

{

int i, j, sum = 0, average;

for (i = 0; i < 4; i++)

{

for (j = 0; j < 5; j++)

{

sum = sum + arr[i, j];

}

average = sum / 5;

Console.WriteLine("Average of city" + (i + 1) + "=" + average);

sum = 0;

}

}

void min\_max(int[,] arr)

{

int i, j, min, max;

for (i = 0; i < 4; i++)

{

j = 0;

max = min = arr[i, j];

for (j = 0; j < 5; j++)

{

if (max < arr[i, j])

max = arr[i, j];

if (min > arr[i, j])

min = arr[i, j];

}

Console.WriteLine("Max of temperature of" + (i+1) + "cities" + max);

Console.WriteLine("Min of temperature of" + (i+1) + "cities" + min);

}

}

staticvoid Main(string[] args)

{

int[,] arr = newint[4, 5] { { 40, 45, 35, 32, 34 }, { 30, 35, 34, 32, 30 }, { 30, 31, 35, 35, 35 }, { 30, 25, 40, 30, 45 } };

Program t1 = newProgram();

t1.temp(arr);

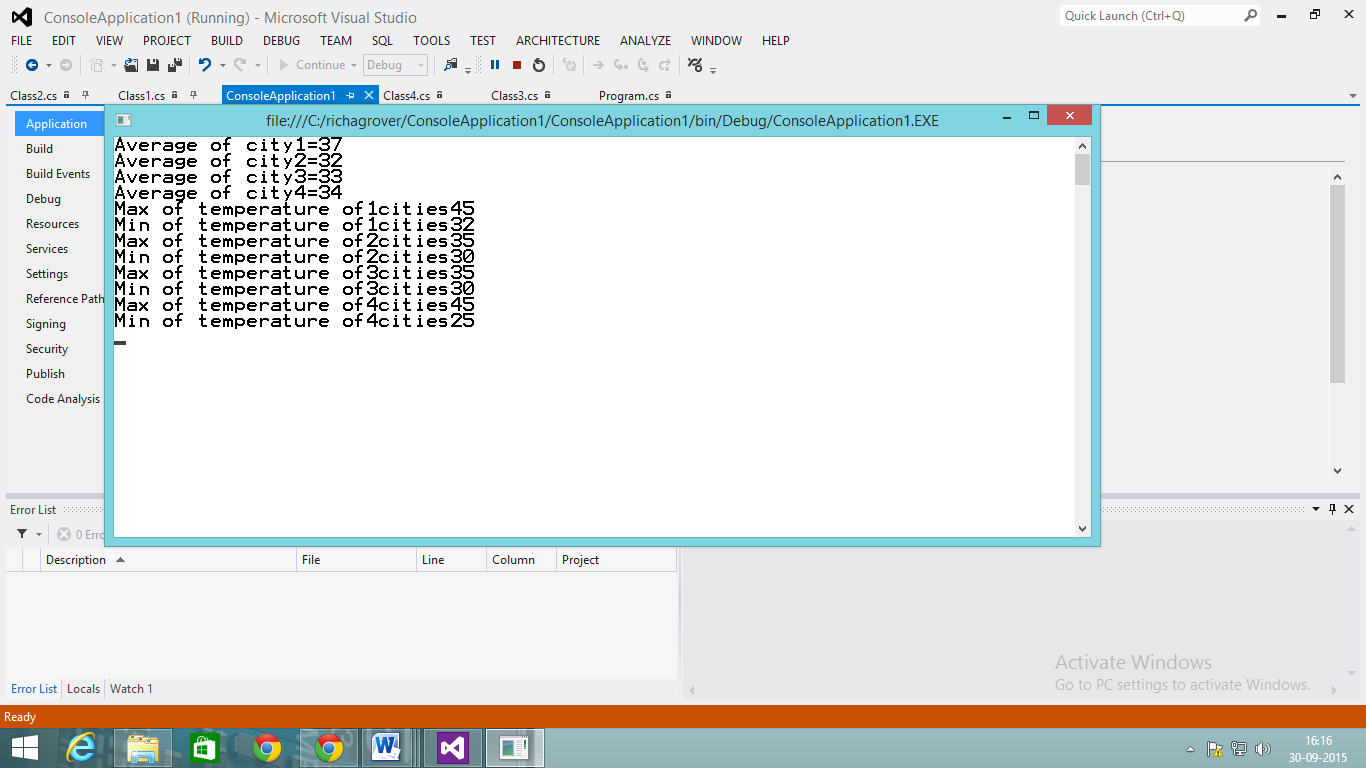
t1.min\_max(arr);

Console.ReadKey();

}

}

}



**Q.8 Write a program that make use of ref , in , out and param key words**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication4

{

classRef\_In\_Out

{

staticvoid Main(string[] args)

{

string name1 = "Akhil";

string name2;

Ref\_In\_Out ob1=newRef\_In\_Out();

Console.WriteLine("\n Name Before Calling RefMethod : "+ name1);

ob1.RefMethod(ref name1);

Console.WriteLine("\n Name After Calling RefMethod : " + name1);

//Console.WriteLine("Name Before Calling OutMethod : " + name2);

ob1.OutMethod(out name2);

Console.WriteLine("\n Name After Calling OutMethod : " + name2);

Console.WriteLine("\n\n Working of in keyword");

foreach (char s in name1)//use of in keyword

{

Console.Write(" "+ s + " ");

}

Console.WriteLine("\n\n Use of param keyword\n");

ob1.UseParams2(1, 2,3,4 );//use of params keyword

Console.ReadLine();

}

privatevoid RefMethod(refstring nameRef)

{

nameRef = "Akhil Mittal";

}

privatevoid OutMethod(outstring nameOut)

{

nameOut = "Akhil Mittal in out method";

//Console.WriteLine(nameOut);

}

publicvoid UseParams2(paramsobject[] list)

{

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

{

Console.Write(" " +list[i] + " ");

}

//Console.WriteLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Practical\_list

{

class Program8

{

staticvoid Main(string[] args)

{

Console.WriteLine("USING param");

using\_param("Hi", "we", "are", "from", "jims");

int par1 = 10, par2 = 20;

Console.WriteLine("\n\nUSING ref");

without\_using\_ref( par1, par2);

Console.WriteLine("Values of a and b (Without using ref)is : {0},{1}",par1,par2);

using\_ref(ref par1, ref par2);

Console.WriteLine("Values of a and b (using ref)is : {0},{1}", par1, par2);

int par3;

Console.WriteLine("\n\nUSING out");

using\_out(out par3);

Console.WriteLine("Square of No. is : {0}", par3);

int par4=786;

Console.WriteLine("\n\nUSING in(By Default)");

using\_in(par4);

Console.ReadKey();

}

staticvoid using\_param(paramsstring[] str)

{

foreach (string s in str)

{

Console.Write(s+" ");

}

}

staticvoid without\_using\_ref(int par1, int par2)

{

int inter;

inter = par1;

par1 = par2;

par2 = inter;

}

staticvoid using\_ref(refint par1,refint par2)

{

int inter;

inter = par1;

par1 = par2;

par2 = inter;

}

staticvoid using\_out(outint par3)

{

par3 = 10;

par3 = par3 \* par3;

}

staticvoid using\_in(int par4)

{

Console.WriteLine("Value of Argument passed is : {0}",par4);

}

}

}

**OUTPUT :**

