Q 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;

using System.Threading.Tasks;

namespace armstrong

{

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ArmstrongNumber

{

class Program

{

static void Main(string[] args)

{

String a;

double c, d, arm = 0;

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

a = Console.ReadLine();

c = a.Length;

int b = int.Parse(a);

Console.WriteLine(c);

int e = 0;

while (e != c)

{

d = b % 10;

b = b / 10;

arm = arm + Math.Pow(d, c);

e = e + 1;

}

if (arm == double.Parse(a))

Console.WriteLine("It is an Armstrong Number");

else

Console.WriteLine("It is not an Armstrong Number");

//Math.Pow()

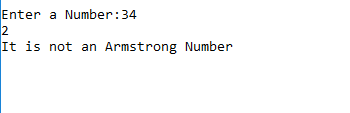
Console.ReadKey();

}

}

}

}



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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static void Main(String[] args)

{

double a, b, c, x1, x2;

Console.WriteLine("Enter a quadratic equation:");

Console.Write("x2:");

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

Console.Write("x:");

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

Console.Write("constant:");

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

Console.WriteLine("Quadratic Equation is: " + a + "x2 + " + b + "x + " + c);

double d = Math.Sqrt(Math.Pow(b, 2) - (4 \* a \* c));

x1 = (-b + d) / (2 \* a);

x2 = (-b - d) / (2 \* a);

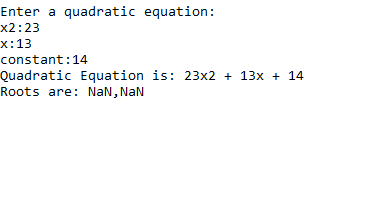
Console.WriteLine("Roots are: " + x1 + "," + x2);

Console.ReadKey();

}

}

}



1. Write a C# Program to implement to Convert Decimal to Binary, Octal , Hexadecimal and reverse too.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace @decimal

{

class Conversion

{

public void Dec2Bin()

{

Console.WriteLine("Decimal to Binary: ");

int num;

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

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

string conv = "";

while (num >= 1)

{

conv += Convert.ToString(num % 2);

num = num / 2;

}

string b = "";

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

{

b = b + conv[i];

}

Console.WriteLine("Binary format: " + b);

}

public void Dec2Oct()

{

Console.WriteLine("\nDecimal to Octal: ");

int num;

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

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

string conv = "";

while (num >= 1)

{

conv += (num % 8).ToString();

num = num / 8;

}

string b = "";

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

{

b = b + conv[i];

}

Console.WriteLine("Octal format: " + b);

}

public void Dec2Hex()

{

int num, a;

Console.WriteLine("\n Decimal to Hexadecimal: \n");

int i = 1, j;

char[] hex = new char[50];

char temp1;

Console.WriteLine("Enter a Decimal Number :");

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

while (num != 0)

{

a = num % 16;

if (a < 10)

a = a + 48;

else

a = a + 55;

temp1 = Convert.ToChar(a);

hex[i++] = temp1;

num = num / 16;

}

Console.Write("HexaDecimal format:");

for (j = i - 1; j > 0; j--)

Console.Write(hex[j]);

}

public void Bin2Dec()

{

int num, decVal = 0, baseVal = 1, r;

Console.Write("Enter Binary Number(1s and 0s): ");

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

int i = 1;

while (num > 0)

{

r = num % 10;

decVal = decVal + r \* baseVal;

num = num / 10;

baseVal = (int)Math.Pow(2, i);

i++;

}

Console.Write("Decimal value: " + decVal);

}

public void Oct2Dec()

{

int num, dec = 0;

int i = 0;

Console.Write("Enter octal number: ");

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

while (num != 0)

{

dec = dec + ((num % 10) \* (int)Math.Pow(8, i));

i++;

num = num / 10;

}

Console.Write("Decimal value: " + dec);

}

public void Hex2Dec()

{

int val = 1, dec = 0;

string Hex;

int i, len, j;

Console.WriteLine("Enter a Hexadecimal Number: ");

Hex = Console.ReadLine();

len = Hex.Length;

j = 1;

for (i = len - 1; i >= 0; i--)

{

switch (Hex[i])

{

case '0':

val = 0;

break;

case '1':

val = 1;

break;

case '2':

val = 2;

break;

case '3':

val = 3;

break;

case '4':

val = 4;

break;

case '5':

val = 5;

break;

case '6':

val = 6;

break;

case '7':

val = 7;

break;

case '8':

val = 8;

break;

case '9':

val = 9;

break;

case 'A':

val = 10;

break;

case 'B':

val = 11;

break;

case 'C':

val = 12;

break;

case 'D':

val = 13;

break;

case 'E':

val = 14;

break;

case 'F':

val = 15;

break;

default:

Console.WriteLine("Not Hexadecimal");

break;

}

dec = dec + val \* (int)Math.Pow(16, j);

j++;

}

Console.WriteLine("Decimal value: " + dec);

}

}

class Program

{

static void Main(string[] args)

{

Conversion cobj = new Conversion();

int choice = 0;

do

{

Console.WriteLine("\n1.Decimal to Binary\n2.Decimal to Octal\n3.Decimal to Hexadecimal");

Console.WriteLine("4.Binary to Decimal\n5.Octal to Decimal\n6.Hexadecimal to Decimal\n7.Exit");

Console.Write("Enter Choice:");

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

switch (choice)

{

case 1:

cobj.Dec2Bin();

break;

case 2:

cobj.Dec2Oct();

break;

case 3:

cobj.Dec2Hex();

break;

case 4:

cobj.Bin2Dec();

break;

case 5:

cobj.Oct2Dec();

break;

case 6:

cobj.Hex2Dec();

break;

case 7:

Console.WriteLine("Exit");

break;

default:

Console.WriteLine("Wrong Choice\n");

break;

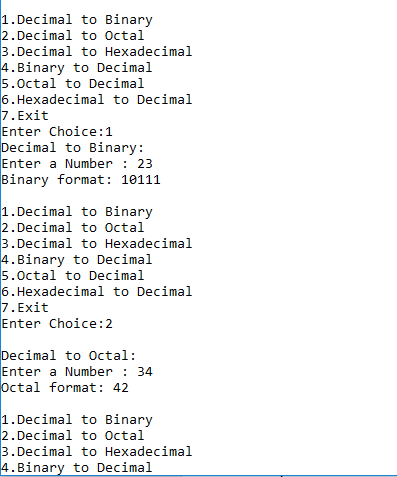
}

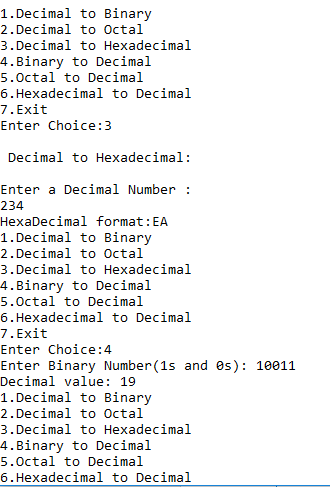
} while (choice != 7);

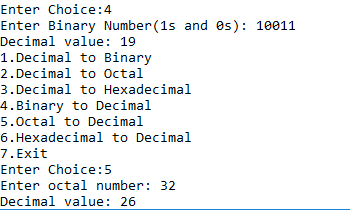
}

}

}







1. Write a C# Program to implement all matrix operation like addition, multiplication, Transpose and sum of diagonal of any resultant matrix.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace matrix

{

class matrix

{

static void display(int[,] c)

{

int i, j;

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

{

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

{

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

}

Console.WriteLine("\n");

}

}

static void add(int[,] a, int[,] b)

{

int[,] c = new int[2, 2];

//add

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

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

{

c[i, j] = a[i, j] + b[i, j];

}

//display

Console.WriteLine("Addition:");

display(c);

}

static void multiply(int[,] a, int[,] b)

{

int i, j, k;

int[,] c = new int[2, 2];

//multiply

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

{

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

{

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

{

c[i, k] = c[i, k] + (a[i, j] \* b[j, k]);

}

}

}

//display

Console.WriteLine("Multiplication:");

display(c);

}

static void Transpose(int[,] a, int[,] b)

{

int[,] c = new int[2, 2];

int i, j;

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

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

{

c[i, j] = a[j, i];

}

Console.WriteLine("Transpose of a:");

display(c);

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

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

{

c[i, j] = b[j, i];

}

Console.WriteLine("Transpose of b:");

display(c);

}

static void Main(String[] args)

{

int[,] a = new int[2, 2];

int[,] b = new int[2, 2];

Console.WriteLine("Enter the Matrices:");

Console.WriteLine("Enter first matrix:");

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

{

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

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

}

Console.WriteLine("Enter second matrix:");

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

{

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

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

}

add(a, b);

multiply(a, b);

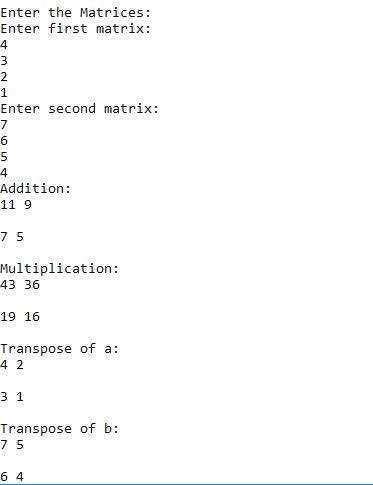
Transpose(a, b);

Console.ReadKey();

}

}

}



7. WAP to store the numbers in an array and search for a particular number and their occurrences in that.(number of times)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace occurrences

{

class Ary

{

public void Search(int[] a)

{

Console.WriteLine("Enter the element you want yto search in the array");

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

//int flag=0;

int count = 0;

for (int i = 0; i < 5; i++)

{

if (a[i] == k)

{

count++;

}

}

if (count == 0)

{

Console.WriteLine("Element Not Found");

}

else

{

Console.WriteLine("Element Found" + count + " times ");

}

}

public static void Main(String[] args)

{

Ary obj = new occurrences.Ary();

int[] a = new int[5];

Console.WriteLine("Enter the elements in Array");

for (int i = 0; i < 5; i++)

{

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

}

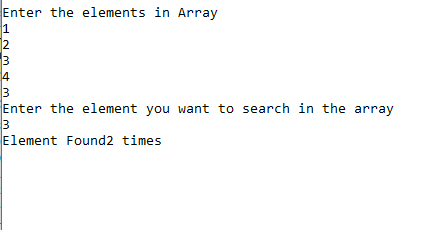
obj.Search(a);

Console.ReadKey();

}

}

}



8. Write a program to implement Jagged  Arrays.(Print any pattern)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace jagged

{

class Class2

{

static void Main(string[] args)

{

int i, j;

char[][] num = new char[6][];

num[0] = new char[1];

num[1] = new char[2];

num[2] = new char[3];

num[3] = new char[4];

num[4] = new char[5];

num[5] = new char[6];

for(i=0;i<6;i++)

{

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

{

num[i][j] = '\*';

Console.Write(num[i][j]);

}

Console.WriteLine();

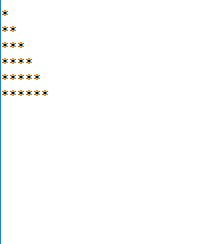
}

Console.ReadKey();

}

}

}



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.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace bank

{

class Bank

{

String name;

long accountNo;

String accType;

int Balance;

public Bank()

{

Console.Write("Enter Name: ");

name = Console.ReadLine();

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

accountNo = Convert.ToInt64(Console.ReadLine());

Console.Write("Enter Account Type: ");

accType = Console.ReadLine();

Console.Write("Enter Balance: ");

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

}

public void Deposit(int deposit\_amnt)

{

Balance = Balance + deposit\_amnt;

Console.WriteLine("\nBalance: " + Balance + "\n");

}

public void Withdraw(int withdraw\_amount)

{

if (accType == "Savings")

{

if (Balance <= 1000)

Console.WriteLine("\nNot appropriate Balance\n");

else

if ((Balance - withdraw\_amount) <= 0)

Console.WriteLine("\nBalance Insufficient\n");

else

Balance = Balance - withdraw\_amount;

Console.WriteLine("Balance: " + Balance);

}

else

{

Balance = Balance - withdraw\_amount;

Console.WriteLine("Balance: " + Balance);

}

}

public void disp\_details()

{

Console.WriteLine("\nName: " + name);

Console.WriteLine("Balance: " + Balance + "\n");

}

}

class Class3

{

static void Main(String[] args)

{

byte ch;

int amt;

Bank bobj = new Bank();

do

{

Console.WriteLine("\n1.Deposit Amount\n2.Withdraw Amount\n3.Display Details\n4.Exit");

Console.Write("Enter Choice: ");

ch = Convert.ToByte(Console.ReadLine());

switch (ch)

{

case 1:

Console.Write("\nEnter Amount: ");

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

bobj.Deposit(amt);

break;

case 2:

Console.WriteLine("\nEnter Amount: ");

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

bobj.Withdraw(amt);

break;

case 3:

bobj.disp\_details();

break;

case 4:

Console.WriteLine("\nExit");

break;

default:

Console.WriteLine("\nWrong Choice\n");

break;

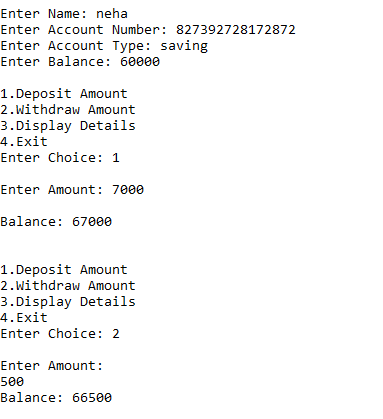
}

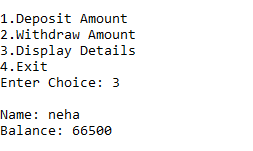
} while (ch != 4);

}

}

}





10. Write a C# Program to find out the area of the triangle , square, and rectangle using method overloading.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace area

{

class Class1

{

int area(int s)

{

return s \* s;

}

void area(int h, int b)

{

float ar;

ar = (float)(0.5 \* h \* b);

Console.WriteLine("Area of triangle: " + ar);

}

int area(int l, int b, int ar)

{

ar = l \* b;

return ar;

}

static void Main(string[] args)

{

Class1 obj = new Class1();

Console.Write("Enter the side of the square: ");

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

Console.WriteLine("Area of sqaure : " + obj.area(a));

Console.Write("Enter height of triangle: ");

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

Console.Write("Enter base of triangle: ");

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

obj.area(h, b);

Console.Write("Enter the length of the rectangle: ");

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

Console.Write("Enter the breadth of the rectangle: ");

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

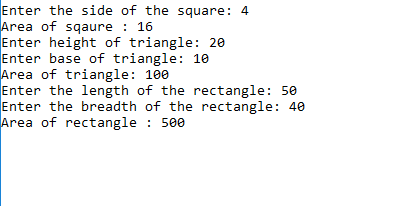
Console.WriteLine("Area of rectangle : " + obj.area(l, b, 0));

Console.ReadLine();

}

}

}



11. Write a C# Program to implement the swapping of two values by call by value and call by reference using struct  and methods.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace swap

{

class swap

{

public void swap\_by\_value(int a, int b)

{

a = a + b;

b = a - b;

a = a - b;

Console.WriteLine("Swapped Values : " + a + "," + b);

}

public void swap\_by\_ref(ref int a, ref int b)

{

a = a + b;

b = a - b;

a = a - b;

Console.WriteLine("Swapped Values : " + a + "," + b);

}

public void swap\_by\_out(out int a, out int b)

{

a = 10;

b = 20;

Console.WriteLine("Values Without swap : " + a + "," + b);

a = a + b;

b = a - b;

a = a - b;

Console.WriteLine("Values after swap : " + a + "," + b);

}

}

class Class2

{

static void Main(string[] args)

{

int a, b;

swap s = new swap();

Console.Write("Enter a: ");

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

Console.Write("Enter b: ");

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

Console.WriteLine("Pass By value:");

Console.WriteLine("Values before swap : " + a + "," + b);

s.swap\_by\_value(a, b);

Console.WriteLine("Values after swap : " + a + "," + b);

Console.WriteLine("Pass By refrence (ref):");

Console.WriteLine("Values before swap : " + a + "," + b);

s.swap\_by\_ref(ref a, ref b);

Console.WriteLine("Values after swap : " + a + "," + b);

Console.WriteLine("Pass By refrence (out):");

s.swap\_by\_out(out a, out b);

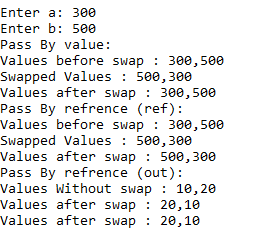
Console.WriteLine("Values after swap : " + a + "," + b);

Console.ReadLine();

}

}

}



Q **Explain the concept of struct.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace @struct

{

public struct myStructure

{

public int num;

public int init

{

get

{

return num;

}

set

{

num = value;

}

}

}

class structProp

{

static void Main(String[] args)

{

myStructure str1 = new myStructure();

Console.WriteLine("enter a value to initialize num: ");

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

str1.init = i;

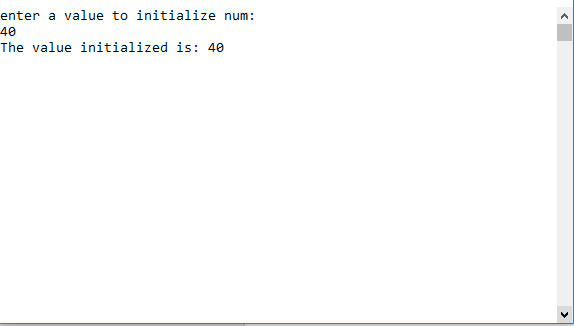
Console.WriteLine("The value initialized is: " + str1.init);

Console.ReadKey();

}

}

}



**Q. Write a C# Program to implement the use of all kind of Inheritance (like: Single level, Multi-level,) .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.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace level

{

public class A

{

public virtual void display()

{

Console.WriteLine("This is Super Base class");

}

}

public class B : A

{

public override void display()

{

base.display();

Console.WriteLine("This is the Base class");

}

}

public class C : B

{

protected new void display()

{

Console.WriteLine("This is the derived class");

}

}

class inherit

{

static void Main(String[] args)

{

A obj1 = new A();

obj1.display();

A obj2 = new B();

obj2.display();

A obj3 = new C();

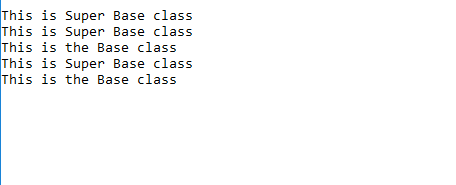
obj3.display();

Console.ReadKey();

}

}

}



Q **Write a used defined  exception handler if the first letter of the given input is not capital and handle all other defied exception handler.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace user\_defined

{

class excHandling

{

static void Main(string[] args)

{

try

{

Console.WriteLine("Enter your name");

string name = Console.ReadLine();

if (name[0] >= 'a' && name[0] <= 'z')

{

throw (new FirstLetterException("Error Raised"));

}

}

catch (FirstLetterException e)

{

Console.WriteLine(e.Message);

}

Console.ReadKey();

}

}

class FirstLetterException : Exception

{

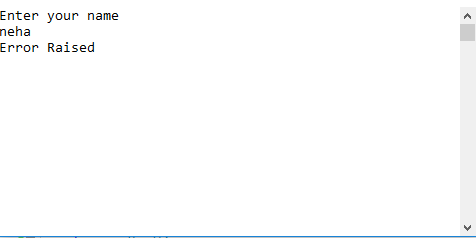
public FirstLetterException(string message) : base(message)

{

}

}

}



QW**rite a C# Program to implement the string operations :- string reverse, concatenate, copy, replace using functions in class.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace copy

{

class Program

{

static void Main()

{

string str1;

str1 = "welcome to c# ";

string str2;

str2 = "first program ";

Console.WriteLine("concatination of two strings is: " + string.Concat(str1, str2));

Console.WriteLine("copy of string: " + string.Copy(str1));

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

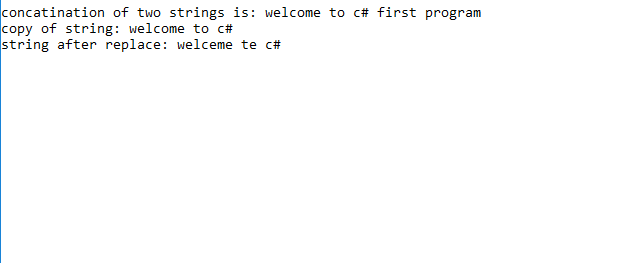
Console.WriteLine("string after replace: " + str1);

Console.ReadKey();

}

}

}



Q **Write a C# Program to implement the Sorting of names in alphabetic order or lexicographical order**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace sort

{

class sort

{

public static void Main()

{

string[] a = new string[4];

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

{

Console.WriteLine("enter string");

a[i] = Console.ReadLine();

}

Array.Sort(a);

Console.WriteLine("\n" + "strings after sorting ");

foreach (string s in a)

{

Console.WriteLine(s);

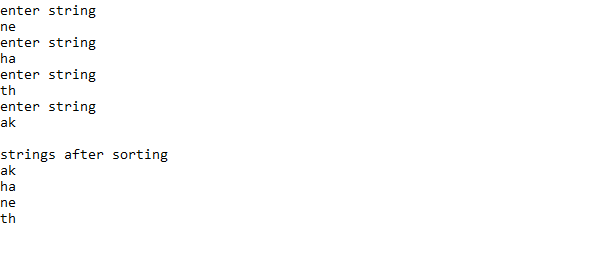
}

Console.ReadKey();

}

}

}



Q **Write a C# program to show multiple inheritance**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace inherit

{

class Shape

{

protected int width;

protected int height;

public void setWidth(int w)

{

width = w;

}

public void setHeight(int h)

{

height = h;

}

}

public interface area

{

int getCost(int area);

}

class Rectangle : Shape, area

{

public int getArea()

{

return (width \* height);

}

public int getCost(int area)

{

return area \* 70;

}

}

class multi

{

static void Main(string[] args)

{

Rectangle Rect = new Rectangle();

int area;

Rect.setWidth(20);

Rect.setHeight(10);

area = Rect.getArea();

Console.WriteLine("Total area: " + area);

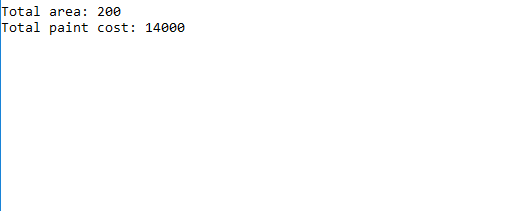
Console.WriteLine("Total paint cost: " + Rect.getCost(area));

Console.ReadKey();

}

}

}



Q **Write a C# Program to call any method that agrees with its signature and return type using delegate.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace @delegate

{

delegate int Num1(int n);

class delegateEx

{

public int AddNum(int p)

{

int num = 10;

num += p;

return num;

}

public static void Main(string[] args)

{

int x, y;

int p;

//create delegate instances

delegateEx del = new delegateEx();

Num1 nc1 = del.AddNum;

p = nc1(15);

Console.WriteLine("Value of Num after adding 15 is: " + p);

p = nc1(10);

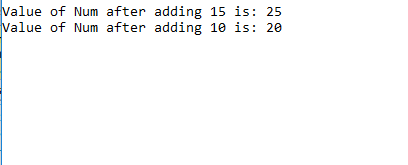
Console.WriteLine("Value of Num after adding 10 is: " + p);

Console.ReadKey();

}

}

}



Q. **Write a C# Program to use of an anonymous method that count from 0 to 10.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace count

{

delegate void Func();

class del1

{

static void Main()

{

Console.WriteLine("Counting from 0 to 10 through anonymous delegate");

Func count = delegate ()

{

for (int i = 0; i <= 10; i++)

Console.Write(i + "\t");

};

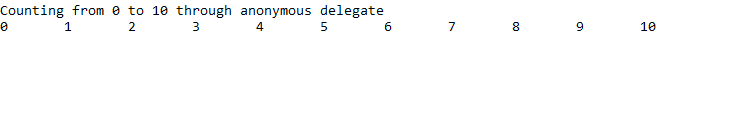
count();

Console.ReadKey();

}

}

}



Q. **One Complete console application that usage delegates and event mechanism to fire, wire and handle an event**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace fire

{

delegate void handler();

class event1

{

public event handler raise;

public void onevent()

{

if (raise != null)

raise();

}

}

class event2

{

public void sec\_hand()

{

Console.WriteLine("event raised by first class");

}

}

class EventHandler

{

static void handler()

{

Console.WriteLine("event raised by second class");

}

public static void Main(string[] args)

{

event1 evt1 = new event1();

event2 evt2 = new event2();

evt1.raise += handler;

evt1.raise += evt2.sec\_hand;

evt1.onevent();

Console.WriteLine("");

Console.WriteLine("after removing event\_handler: ");

evt1.raise -= evt2.sec\_hand;

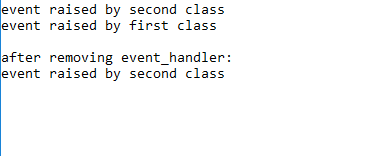
evt1.onevent();

Console.ReadKey();

}

}

}



Q. **Write a C# Program to create a statement lambda that returns the factorial of the value it is passed.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

delegate int del(int i);

namespace lambda

{

class lemStat

{

static del Fact =

(n) =>

{

if (n <= 1)

return 1;

else

return n \* Fact(n - 1);

};

static void Main(string[] args)

{

Console.WriteLine("enter a number");

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

int l = lemStat.Fact(i);

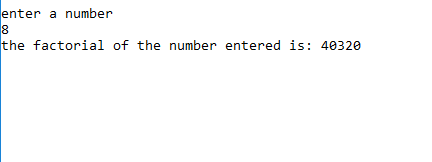
Console.WriteLine("the factorial of the number entered is: " + l);

Console.ReadLine();

}

}

}



Q