Assignment-2

1. Print sum and average of elements of an Array

using System;

namespace sum\_and\_average

{

class Program

{

static void Main(string[] args)

{

int i, sum = 0;

double Average;

Console.WriteLine("Enter numbers one by one");

for (i=0;i<=20;i++)

{

sum = sum + i;

}

Average = sum / 20;

Console.WriteLine("The sum of 20 natural numbers is" + sum);

Console.WriteLine("The Aversge of 20 natural numbers is " + Average);

}

}

}

2. WAP to display sum of 3 by 3 matrices

using System;

namespace Matrixsum

{

public class Program

{

public static void Main()

{

int i, j, n;

int[,] arr1 = new int[20, 20];

int[,] arr2 = new int[20, 20];

int[,] arr3 = new int[20, 20];

// setting matrix row and columns size

n = 3;

Console.Write("Enter elements in the first matrix:\n");

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

{

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

{

arr1[i, j] = Convert.ToInt32(Console.ReadLine());

}

}

Console.Write("Enter elements in the second matrix:\n");

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

{

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

{

arr2[i, j] = Convert.ToInt32(Console.ReadLine());

}

}

Console.Write("\nFirst matrix is:\n");

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

{

Console.Write("\n");

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

Console.Write("{0}\t", arr1[i, j]);

}

Console.Write("\nSecond matrix is:\n");

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

{

Console.Write("\n");

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

Console.Write("{0}\t", arr2[i, j]);

}

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

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

arr3[i, j] = arr1[i, j] + arr2[i, j];

Console.Write("\nAdding two matrices: \n");

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

{

Console.Write("\n");

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

Console.Write("{0}\t", arr3[i, j]);

}

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

}

}

}

3. WAP to find the maximum element of an integer Array

using System;

namespace Maxelement

{

class Program

{

static void Main(string[] args)

{

int i, n;

int max = 0;

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

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

int[] arr1 = new int[n];

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

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

{

arr1[i] = Convert.ToInt32(Console.ReadLine());

}

max = arr1[0];

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

{

if(arr1[i]>max)

{

max = arr1[i];

}

}

Console.Write("Maximum element is: {0}\n", max);

}

}

}

4. WAP to print column wise sum of elements of 2 D Array

using System;

namespace Coloumnwisesum

{

class Program

{

static void Main(string[] args)

{

int n = 4, m = 4, sum = 0;

int[,] arr1 = new int[m, n];

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

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

{

for (int j = 0; j < 4; ++j)

{

arr1[i, j] = Convert.ToInt32(Console.ReadLine());

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

}

}

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

{

for (int j = 0; j < 4; ++j)

{

Console.WriteLine("{0}\t", + arr1[i, j]);

sum = sum + arr1[j, i];

}

}

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

Console.WriteLine("Sum of the column " +sum);

}

}

}

5. WAP to print row wise sum of elements of 2 D Array

using System;

namespace Rowwisesum

{

class Program

{

static void Main(string[] args)

{

int n = 4, m = 4, sum = 0;

int[,] arr1 = new int[m, n];

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

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

{

for (int j = 0; j < 4; ++j)

{

arr1[i, j] = Convert.ToInt32(Console.ReadLine());

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

}

}

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

{

for (int j = 0; j < 4; ++j)

{

Console.WriteLine("{0}\t", +arr1[i, j]);

sum = sum + arr1[i, j];

}

}

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

Console.WriteLine("Sum of the row " + sum);

}

}

}

ASSIGNMENT-3

Write a program to insert elements in an integer array. ELements should be in order

using System;

namespace InsertArray

{

class Program

{

static void Main(string[] args)

{

int i = 0, pos = 0, item = 0,n;

int[] arr = new int[30];

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

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

Console.WriteLine("Enter the elements");

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

{

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

}

Console.WriteLine("Enter the position to insert:");

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

Console.WriteLine("Enter the new item:");

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

for(i=n;i>=pos;i--){

arr[i] = arr[i - 1];

}

arr[pos - 1] = item;

Console.WriteLine("Array after insertion:");

for (i = 0; i< n+1; i++)

{

Console.WriteLine(" " + arr[i]);

}

Console.WriteLine();

}

}

}

Write a program to delete elements from an integer array.

using System;

using System.Collections;

namespace RemoveElements

{

class Program

{

static void Main(string[] args)

{

ArrayList myList = new ArrayList(10);

myList.Add("10");

myList.Add("20");

myList.Add("A");

myList.Add("Madhuri");

myList.Add("50");

Console.WriteLine("The elements in ArrayList are:");

foreach (var x in myList)

Console.WriteLine(x);

myList.RemoveAt(4);

Console.WriteLine("The elements in arraylist after deletion:");

foreach (var x in myList)

Console.WriteLine(x);

}

}

}

Write a program in C# Sharp to count a total number of duplicate elements in an array

using System;

namespace FindDuplicateelements

{

class Program

{

static void Main(string[] args)

{

int i, n, count=0;

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

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

int[] arr = new int[n];

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

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

{

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

}

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

{

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

{

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

{

count++;

break;

}

}

}

Console.WriteLine("The duplicate elements found in the array are:" + count);

}

}

}

Elegibility Criteria:

using System;

namespace Eligibilitycriteria

{

class Program

{

static void Main(string[] args)

{

int phy, maths, chem;

Console.WriteLine("Input the marks obtained in physics :");

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

Console.WriteLine("Input the marks obtained in chemistry :");

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

Console.WriteLine("Input the marks obtained in Mathematics :");

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

Console.Write("Total marks of Maths,Physics and Chemistry :{0}\n", +(maths+phy+chem));

Console.Write("Total marks of Maths and Physics :{0}\n", +(maths+phy));

if (maths >= 65 || phy >= 55 || chem >= 50 || (maths + phy + chem) >= 180 || (maths+ phy) >= 140)

{

Console.WriteLine("The candidate is eligible for admission.");

}

else

{

Console.WriteLine("The candidate is not eligible for admission.");

}

}

}

}

Electriccity Bill:

using System;

namespace ElectriccityBill

{

class Program

{

static void Main(string[] args)

{

int custid, conunit;

double charge, surcha = 0, amt, netchr;

string name;

Console.WriteLine("Customer IDNO:");

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

Console.WriteLine("Customer Name:");

name = Console.ReadLine();

Console.WriteLine("Unit Consumed :");

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

if (conunit < 200)

charge = 1.20;

else if (conunit >= 200 && conunit < 400)

charge = 1.50;

else if (conunit >= 400 && conunit < 600)

charge = 1.80;

else

charge = 2.00;

amt = conunit \* charge;

if (amt > 300)

surcha = amt \* 15 / 100.0;

netchr = amt + surcha;

if (netchr < 100)

netchr = 100;

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

Console.Write("Customer IDNO :{0}\n", custid);

Console.Write("Customer Name :{0}\n", name);

Console.Write("unit Consumed :{0}\n", conunit);

Console.Write("Amount Charges @Rs. {0} per unit :{1}\n", charge, amt);

Console.Write("Surchage Amount :{0}\n", surcha);

Console.Write("Net Amount Paid By the Customer :{0}\n", netchr);

}

}

}

Assignment-8

using System**;**

namespace Animalsound

{

abstract class Animal

{

public abstract void animalSound();

}

class Cat : Animal

{

public override void animalSound()

{

Console.WriteLine("The cats meow");

}

}

class Dog : Animal

{

public override void animalSound()

{

Console.WriteLine("The dogs bark");

}

}

class Program

{

static void Main(string[] args)

{

Cat myCat = new Cat();

myCat.animalSound();

Dog myDog = new Dog();

myDog.animalSound();

}

}

}

Abstract classes

using System;

namespace ConsoleApp3

{

abstract class A

{

protected A()

{

Console.WriteLine("This is Constructor of abstract class");

}

public void Method()

{

Console.WriteLine("This is a normal method of abstract class");

}

public abstract void a\_method();

}

class Subclass : A

{

public override void a\_method()

{

Console.WriteLine("This is abstract method");

}

}

class Program

{

static void Main(string[] args)

{

Subclass sc = new Subclass();

sc.Method();

sc.a\_method();

}

}

}