**1. Describe the enumerations programming constructs, which provides a human-readable form of a series of related constant values in C#.**

using System;

namespace ProgramOne

{

class Program

{

enum CollegeDays

{

MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY

}

static void Main(string[] args)

{

foreach (var day in Enum.GetValues(typeof(CollegeDays)))

{

Console.WriteLine("{0} : {1}",day, (int) day);

}

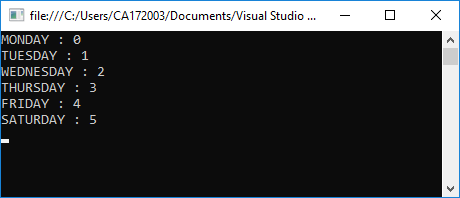
Console.Read();

}

}

}

**OUTPUT:**



**3. Check Whether the Entered Year is a Leap Year or No.**

using System;

namespace ProgramThree

{

class Program

{

static void Main(string[] args)

{

long year;

Console.WriteLine("Enter the year");

year = Int64.Parse(Console.ReadLine());

if (checkYear(year))

Console.WriteLine("{0} is a Leap year", year);

else

Console.WriteLine("{0} is not a Leap year", year);

Console.ReadLine();

}

static bool checkYear(long year) {

if (year % 400 == 0)

return true;

else if (year % 100 == 0)

return false;

else if (year % 4 == 0)

return true;

else

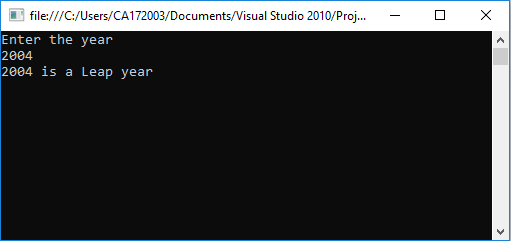
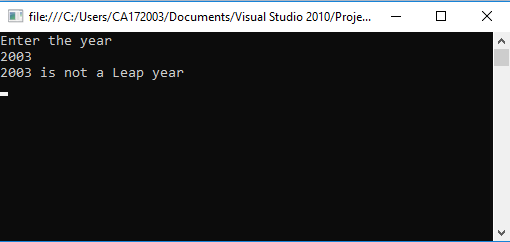
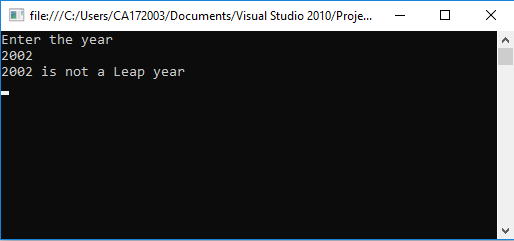
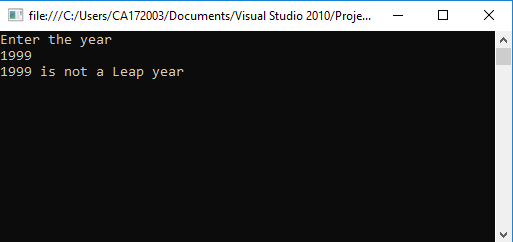
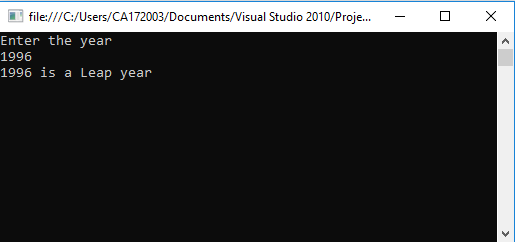
return false;

}

}

}

**OUTPUT:**



**5. Program to display the addition, subtraction, multiplication and division of two number using console applications.**

using System;

namespace ProgramFive

{

class Program

{

static void Main(string[] args)

{

double num1, num2;

double sum, sub, mul, div;

Console.WriteLine("Enter the two numbers");

num1 = Double.Parse(Console.ReadLine());

num2 = Double.Parse(Console.ReadLine());

sum = num1 + num2;

sub = num1 - num2;

mul = num1 \* num2;

div = num1 / num2;

Console.WriteLine("Addition: {0}", sum);

Console.WriteLine("Substraction: {0}", sub);

Console.WriteLine("Multiplication: {0}", mul);

Console.WriteLine("Division: {0}", div);

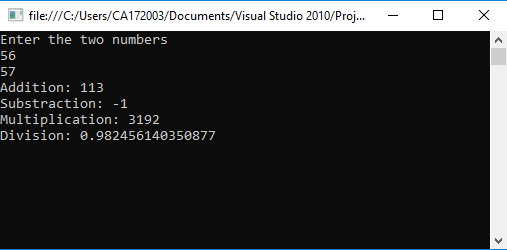
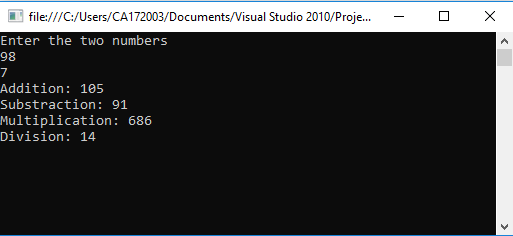
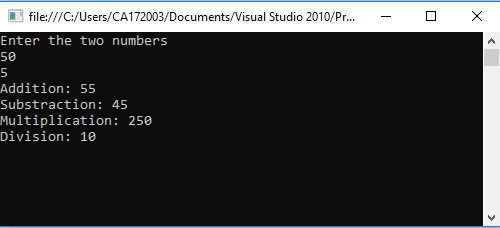
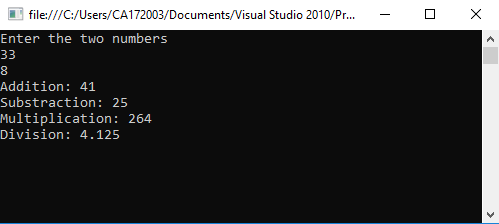
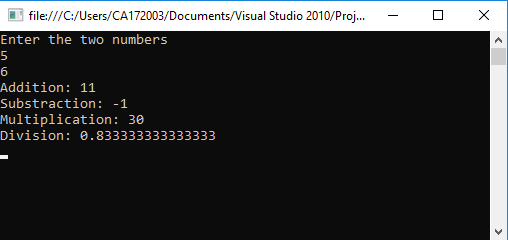
Console.ReadLine();

}

}

}

**OUTPUT:**



**6. Program to display the first 10 natural numbers and their sum using console application.**

using System;

namespace ProgramSix

{

class Program

{

static void Main(string[] args)

{

int sum = 0;

Console.WriteLine("First 10 natural numbers");

for (int i = 1; i <= 10; i++)

{

sum += i;

Console.WriteLine(i);

}

Console.WriteLine("Sum: {0}", sum);

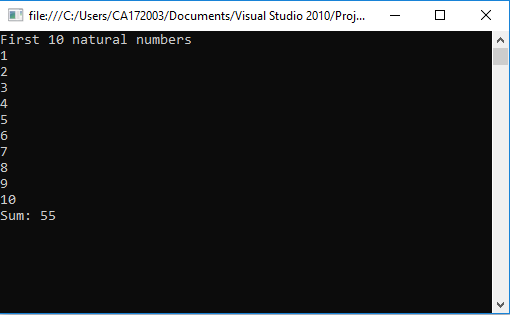
Console.ReadLine();

}

}

}

**OUTPUT:**



**7. Program to display the addition using the windows application.**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace ProgramSeven

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

int num1 = Int16.Parse(textBox1.Text);

int num2 = Int16.Parse(textBox2.Text);

int sum = num1 + num2;

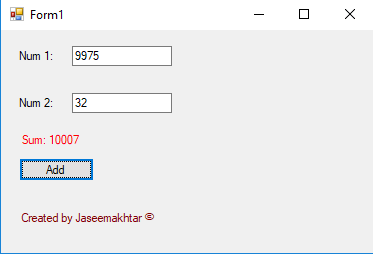
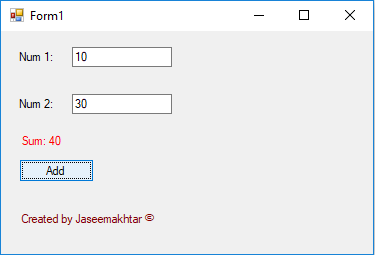
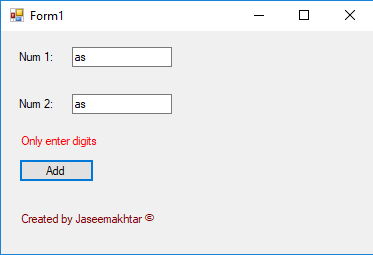
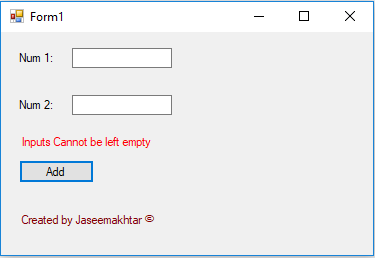
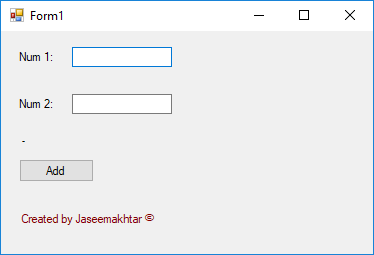
label1.Text = "Sum: " + sum;

}

}

}

**OUTPUT:**



**9. Write a program to convert input string from lower to upper and upper to lower case.**

using System;

namespace ProgramNine

{

class Program

{

static void Main(string[] args)

{

string input;

Console.WriteLine("Enter any word / sentence");

input = Console.ReadLine();

Console.WriteLine("Upper case: {0}", input.ToUpper());

Console.WriteLine("Lower case: {0}", input.ToLower());

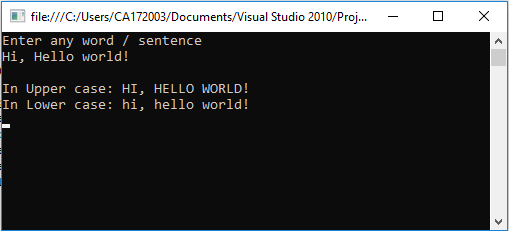
Console.ReadLine();

}

}

}

**OUTPUT:**



**12. Perform operator overloading.**

using System;

namespace ProgramTwelve

{

class Rectangle

{

int width;

int height;

Rectangle(int width, int height) {

this.width = width;

this.height = height;

}

public static Rectangle operator +(Rectangle a, Rectangle b)

{

int totalWidth = a.width + b.width;

int totalHeight = a.height + b.height;

return new Rectangle(totalWidth, totalHeight);

}

static void Main(string[] args)

{

Rectangle r1 = new Rectangle(60, 40);

Rectangle r2 = new Rectangle(80, 50);

Console.WriteLine("\*\*\*\*\*\*\*\*\*Operator Overloading\*\*\*\*\*\*\*\*\*\*\*\*\n");

Console.WriteLine("Rectangle(R1)");

Console.WriteLine("Width: {0}", r1.width);

Console.WriteLine("Height: {0}", r1.height);

Console.WriteLine();

Console.WriteLine("Rectangle(R2)");

Console.WriteLine("Width: {0}", r2.width);

Console.WriteLine("Height: {0}", r2.height);

Console.WriteLine();

Rectangle r3 = r1 + r2;

Console.WriteLine("Total Width (R1 + R2): {0}", r3.width);

Console.WriteLine("Total Height (R1 + R2): {0}", r3.height);

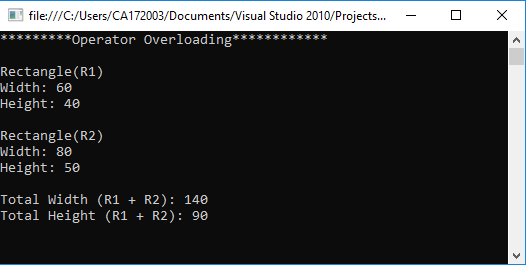
Console.Read();

}

}

}

**OUTPUT:**



**14. Find the second largest element in a single dimensional array.**

using System;

namespace ProgramFourteen

{

class Program

{

static void Main(string[] args)

{

int n;

Console.WriteLine("Enter the size of the array");

n = Int16.Parse(Console.ReadLine());

int[] array = new int[n];

Console.WriteLine("Enter {0} elements into array", n);

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

array[i] = Int16.Parse(Console.ReadLine());

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

int max = array[i];

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

if (array[j] > max) {

int t = array[j];

array[j] = array[i];

array[i] = t;

}

}

}

Console.WriteLine("Second largest element: {0}", array[n - 2]);

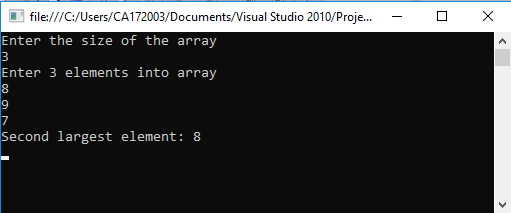
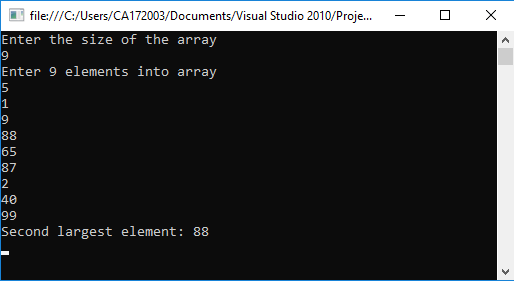
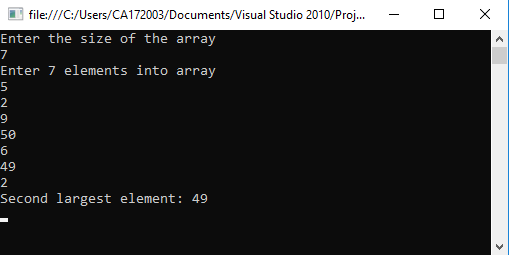
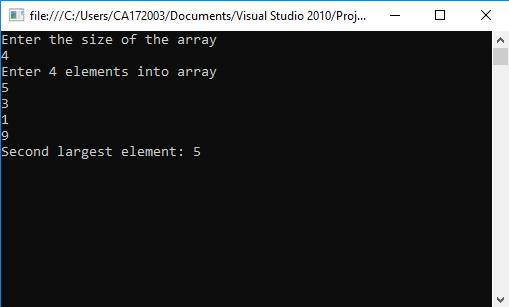
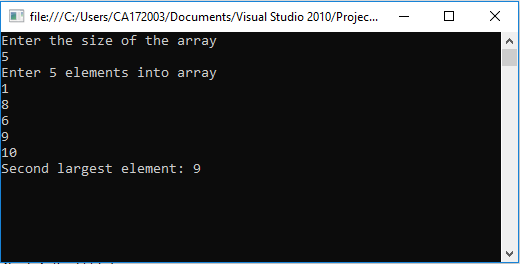
Console.ReadLine();

}

}

}

**OUTPUT:**



**15. Program to illustrate the use of different properties in C#.**

using System;

namespace ProgramFifteen

{

class PropertiesDemo

{

private string name;

private int age;

public string Name{

set {

name = value;

}

get {

return name;

}

}

public int Age {

set {

if (value > 0)

age = value;

}

get {

return age;

}

}

static void Main(string[] args)

{

PropertiesDemo p = new PropertiesDemo();

p.Name = "Jaseem";

p.Age = 23;

PropertiesDemo d = new PropertiesDemo();

d.Name = "Mr.X";

d.Age = -1;

Console.WriteLine("\*\*\*\*\* Properties (Getter & Setter) \*\*\*\*\*");

Console.WriteLine(" p.Name = \"Jaseem\" ");

Console.WriteLine(" p.Age = 23 ");

Console.WriteLine();

Console.WriteLine(" d.Name = \"Mr.X\" ");

Console.WriteLine(" d.Age = -1 ");

Console.WriteLine();

Console.WriteLine(" {0} : {1}", p.Name, p.Age);

Console.WriteLine(" {0} : {1}", d.Name, d.Age);

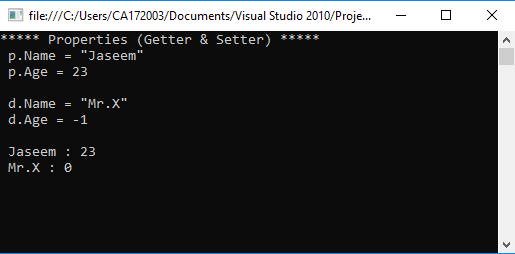
Console.ReadLine();

}

}

}

**OUTPUT:**



**16. Demonstrate Command line arguments processing.**

using System;

namespace ProgramSixteen

{

class Program

{

static void Main(string[] args)

{

if (args.Length >= 2)

{

int num1 = Int32.Parse(args[0]);

int num2 = Int32.Parse(args[1]);

Console.WriteLine("Addition of Command Line Arguments ({0}, {1})", num1, num2);

int sum = num1 + num2;

Console.WriteLine("{0}", sum);

}

else

{

Console.WriteLine("No command line arguments to process");

}

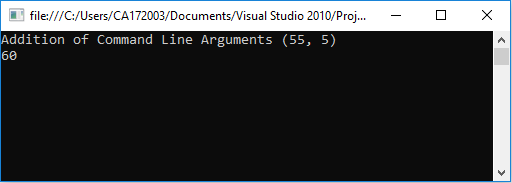
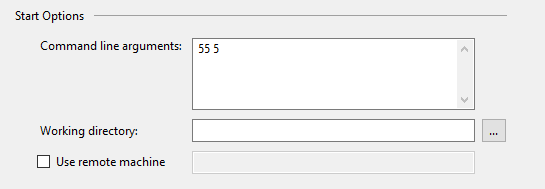
Console.ReadLine();

}

}

}

**OUTPUT:**



**18. Demonstrate Use of Virtual and override keyword in C# with a simple Program.**

using System;

class Animal

{

public virtual void speak() { }

}

class Dog : Animal

{

public override void speak()

{

Console.WriteLine("Woof woof...");

}

}

class Cat : Animal

{

public override void speak()

{

Console.WriteLine("Meow meow...");

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine("#Virual & Override keywords are used in method overriding context\n");

Animal dog = new Dog();

Animal cat = new Cat();

Console.WriteLine("\nDog object:");

dog.speak();

Console.WriteLine("\nCat object:");

cat.speak();

Console.ReadLine();

}

}

**OUTPUT:**

