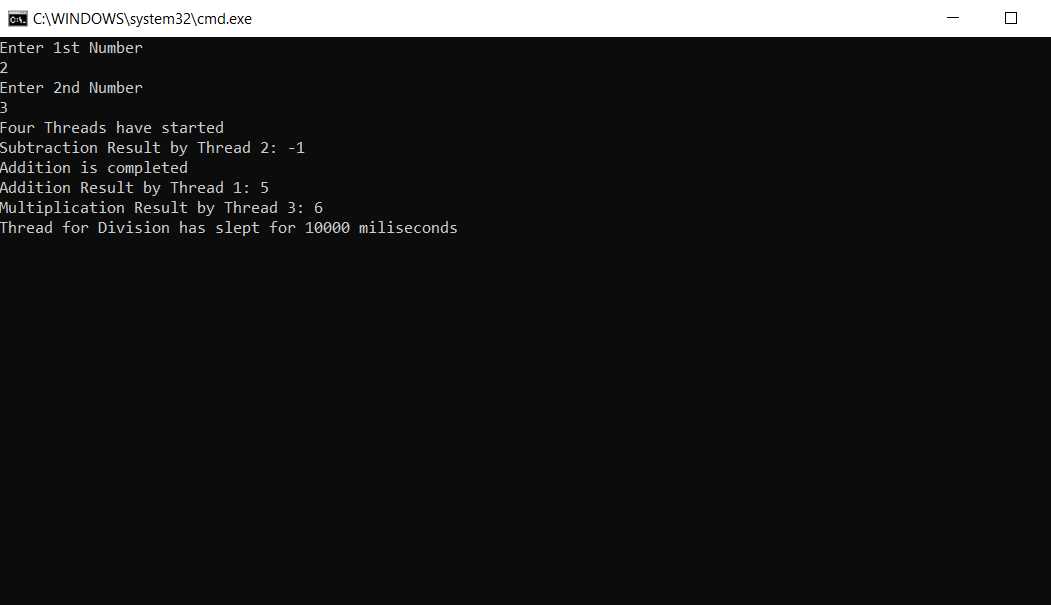
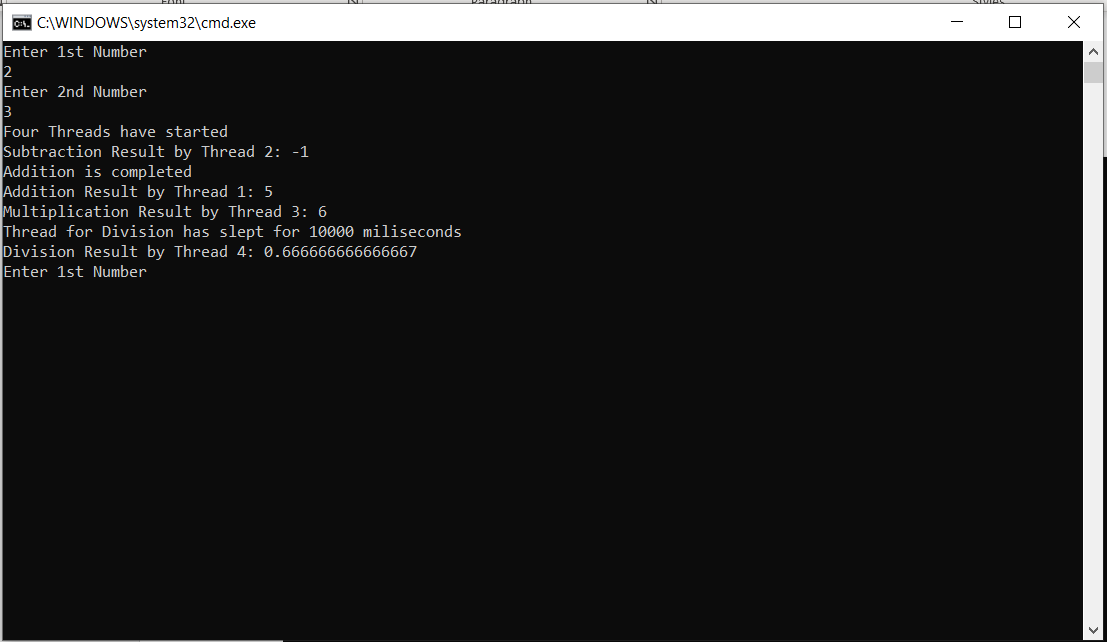
Assignment:3

Aksheshkumar Chaudhari(301087120)

**Output:**



Thread 4 slept for 9000 milli-second and because and because of thread join enter 1st number is asked only after Division method’s thread-4.



**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace Assignment3

{

class Program

{

public double a, b, result;

static void Main(string[] args)

{

Program p = new Program();

p.call();

}

public void Addition()

{

if (a < 0 || b < 0)

{

Console.WriteLine("Thread for Addition has slept for 10000 miliseconds");

Thread.Sleep(10000);

result = a + b;

Console.WriteLine("Addition Result by Thread 1: " + result);

}

else

{

result = a + b;

Console.WriteLine("Addition Result by Thread 1: " + result);

}

}

public void Subtraction()

{

result = a - b;

Console.WriteLine("Subtraction Result by Thread 2: " + result);

}

public void Multiplication()

{

result = a \* b;

Console.WriteLine("Multiplication Result by Thread 3: " + result);

}

public void Division()

{

if (a > 0)

{

Console.WriteLine("Thread for Division has slept for 10000 miliseconds");

Thread.Sleep(9000);

result = a / b;

Console.WriteLine("Division Result by Thread 4: " + result);

}

}

public void message()

{

Program program = new Program();

Console.WriteLine("Enter 1st Number");

program.a = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Enter 2nd Number");

program.b = Convert.ToDouble(Console.ReadLine());

}

public void call()

{

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

{

Program program = new Program();

Console.WriteLine("Enter 1st Number");

program.a = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Enter 2nd Number");

program.b = Convert.ToDouble(Console.ReadLine());

Thread t1 = new Thread(new ThreadStart(program.Subtraction));

Thread t2 = new Thread(new ThreadStart(program.Addition));

Thread t3 = new Thread(new ThreadStart(program.Multiplication));

Thread t4 = new Thread(new ThreadStart(program.Division));

Thread t5 = new Thread(new ThreadStart(program.message));

Console.WriteLine("Four Threads have started");

t1.Start();

t1.Join();

Console.WriteLine(t1.IsAlive ? "Addition will take 5 seconds" : "Addition is completed");

t2.Start();

t3.Start();

t4.Start();

t4.Join();

t5.Start();

t5.Join();

}

}

}

}