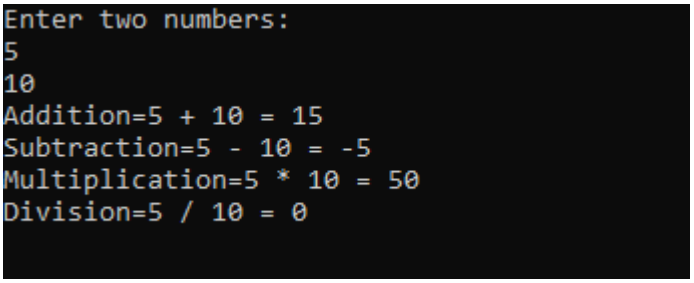
Practical-2
Console-Based Applications using C#

1. W.A.P. to display the addition, subtraction, multiplication, and division of two numbers using a console application.

```
using System;


namespace Labwork2
{

    class pro1
    {
        static void Main(string[] args)
        {
            int num1, num2;
            Console.WriteLine("Enter two numbers:");
            num1 = Convert.ToInt32(Console.ReadLine());
            num2 = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Addition=" + num1 + " + " + num2 + " = " + (num1 +
num2));
            Console.WriteLine("Subtraction=" + num1 + " - " + num2 + " = " + (num1 -
num2));
            Console.WriteLine("Multiplication=" + num1 + " * " + num2 + " = " + (num1 *
num2));
            Console.WriteLine("Division=" + num1 + " / " + num2 + " = " + (num1 /
num2));
            Console.ReadLine();

        }
    }
}
```

Practical-2
Console-Based Applications using C#

- **Output**

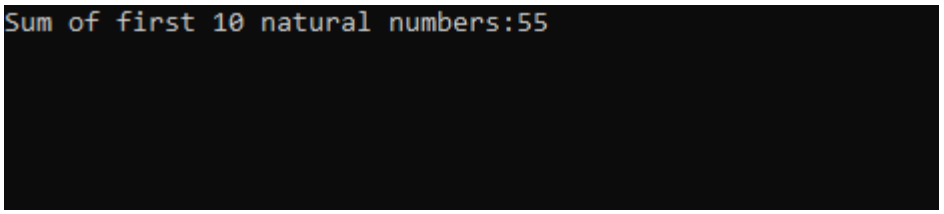


2. W.A.P to display the first 10 natural numbers and their sum using console application.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Labwork2
{
   class pro2
   {

    static void Main(string[] args)
      {
         int sum = 0;

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

         }
         Console.WriteLine("Sum of first 10 natural numbers:" + sum);
```

```
            Console.ReadLine();
        }
    }
}
```

- **Output**

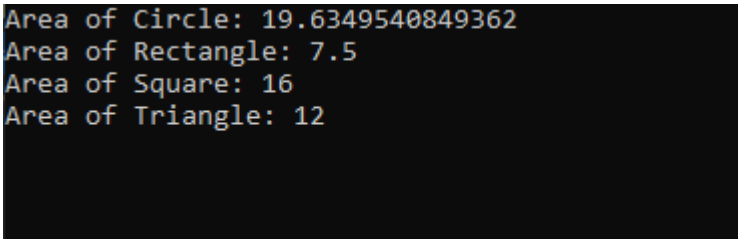


3. W.A.P to calculate area of Circle, Rectangle, Square and Triangle. Which contain two classes in which 1st class contains main method & 2nd class which contains methods to find area for diff. shapes using methodoverloading.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Labwork2
{
    public class ShapeCalculator
    {
        public double CalculateArea(double radius)
        {
            return Math.PI * radius * radius;
        }

        public double CalculateArea(double height, double width)
        {
            return height * width;
        }

        public int CalculateArea(int side)
```

```
    {
        return side * side;
    }

    public double CalculateArea(float tbase, double height)
    {
        return 0.5 * tbase * height;
    }
}
class pro3
{
    static void Main(string[] args)
    {
        ShapeCalculator calculator = new ShapeCalculator();
        double circleArea = calculator.CalculateArea(2.5);
        Console.WriteLine("Area of Circle: " + circleArea);
        double rectangleArea = calculator.CalculateArea(5, 3);
        Console.WriteLine("Area of Rectangle: " + rectangleArea);
        int squareArea = calculator.CalculateArea(4);
        Console.WriteLine("Area of Square: " + squareArea);
        double triangleArea = calculator.CalculateArea(6f, 4);
        Console.WriteLine("Area of Triangle: " + triangleArea);

        Console.ReadLine();

    }

}
}
```

- **Output**



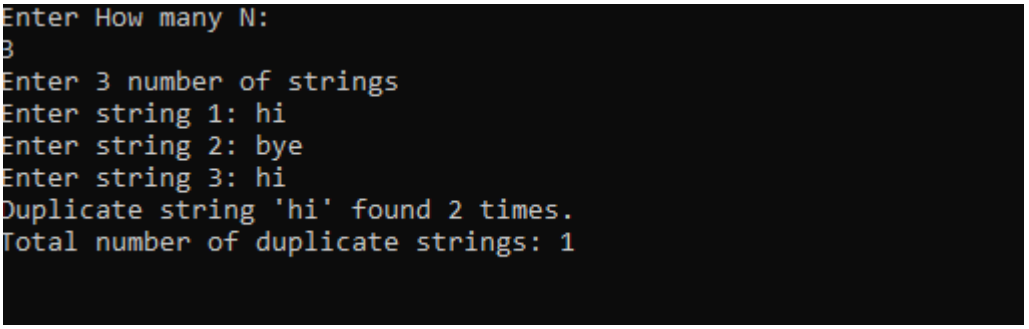
Practical-2
Console-Based Applications using C#

4. W.A.P to get n number of strings from the user. Find out total no. of duplicate strings and display duplicate strings along with duplicate occurrence using 1D array.

```
using System;

namespace Labwork2
{
   class pro4
   {
      public static void Main(string[] args)
      {
         int n = 0;
         int count = 0;
         Console.WriteLine("Enter How many N:");
         n = Convert.ToInt32(Console.ReadLine());
         Console.WriteLine("Enter " + n + " number of strings");
         string[] strarray = new string[n];

         for (int i = 0; i < n; i++)
         {
            Console.Write($"Enter string {i + 1}: ");
            strarray[i] = Console.ReadLine();
         }

         for (int i = 0; i < n; i++)
         {
            int duplicateCount = 0;
            for (int j = i + 1; j < n; j++)
            {
               if (strarray[i] == strarray[j])
               {
                  duplicateCount++;
               }
            }

            if (duplicateCount > 0)
            {
               count++;
```

```
                Console.WriteLine("Duplicate string '" + strarray[i] + "' found " +
(duplicateCount + 1) + " times.");
            }
        }

        Console.WriteLine("Total number of duplicate strings: " + count);
        Console.ReadLine();
    }
  }
}
```

- **Output**



5. W.A.P. to find max and min number from an integer array. Create a method getMinMax() by passing out parameter.

```
using System;

namespace Labwork2
{
   class pro5
   {
      static void Main(string[] args)
      {
         int[] intarr = new int[5] { 2, 10, 12, 6, 8};
         int n = intarr.Length;
         Console.WriteLine("Array Elements:");
         for(int i=0;i<n;i++)
         {
            Console.WriteLine(intarr[i] + " ");
         }
```

```
            getMinMAx(intarr, n);
            Console.ReadLine();

        }
        public  static void getMinMAx(int[] intarr,int n)
        {
            int MAX = intarr[0];
            int MIN = intarr[0];
            for(int i=0;i<n;i++)
            {
              if(MAX < intarr[i])
               {
                  MAX = intarr[i];
               }
              else if(MIN > intarr[i])
               {
                  MIN = intarr[i];
               }
            }
            Console.WriteLine("MAX value: " + MAX);
            Console.WriteLine("MIN value: " + MIN);
        }
    }
}
```

- **Output**

Practical-2
Console-Based Applications using C#

6. Write a Program to create an int Jagged Array which consists of at least 5 arrays in it. Sort every array of Jagged array and display all jagged arrays after sorting.

```
using System;
namespace Labwork2
{
   class pro6
   {
      static void Main(string[] args)
      {
         int[][] jaggedArray = new int[5][];
         jaggedArray[0] = new int[] { 50, 21, 90, 1, 3 };
         jaggedArray[1] = new int[] { 82, 44, 71 };
         jaggedArray[2] = new int[] { 60, 3, 2, 93 };
         jaggedArray[3] = new int[] { 1, 7, 2, 44 };
         jaggedArray[4] = new int[] { 10, 4, 66, 2, 30 };
         for (int i = 0; i < jaggedArray.Length; i++)
         {
            Array.Sort(jaggedArray[i]);
         }
         Console.WriteLine("Sorted Jagged Arrays:");
         for (int i = 0; i < jaggedArray.Length; i++)
         {
             Console.Write("Array " + (i + 1) + ": ");
            for (int j = 0; j < jaggedArray[i].Length; j++)
            {
               Console.Write(jaggedArray[i][j] + " ");
            }
           Console.WriteLine(); }

  Console.ReadLine()} }}
```

- **Output**