PRACTICAL - 2

Console Based Applications using C#

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

Source Code :

```
using System;
namespace BasicCalculator
{
   class Program
   {
      static void Main(string[] args)
       {
          Console.Write("Enter the first number: ");
          double num1 = Convert.ToDouble(Console.ReadLine());

          Console.Write("Enter the second number: ");
          double num2 = Convert.ToDouble(Console.ReadLine());

          double addition = num1 + num2;
          double subtraction = num1 - num2;
          double multiplication = num1 * num2;
          double division = num1 / num2;

          Console.WriteLine($"Addition: {num1} + {num2} = {addition}");
          Console.WriteLine($"Subtraction: {num1} - {num2} = {subtraction}");
          Console.WriteLine($"Multiplication: {num1} * {num2} = {multiplication}");
```

PRACTICAL - 2

```
            Console.WriteLine($"Division: {num1} / {num2} = {division}");
            Console.ReadLine();
        }
    }
}
```

OutPut :

Enter the first number: 100

Enter the second number: 50

Addition: 100 + 50 = 150

Subtraction: 100 - 50 = 50

Multiplication: 100 * 50 = 5000

Division: 100 / 50 = 2

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

Source Code :

```
using System;

namespace FirstTenNaturalNumbers
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("First 10 Natural Numbers and their Sum");
```

PRACTICAL - 2

```
        Console.WriteLine(" ---------------------------------------- ");

        int n = 10;
        int sum = 0;

        for (int i = 1; i <= n; i++)
        {
           Console.Write(i + " ");
           sum += i;
        }
        Console.WriteLine();
        Console.WriteLine($"Sum of first 10 natural numbers: {sum}");
        Console.ReadLine();
      }
   }
}
```

OutPut :

First 10 Natural Numbers and their Sum

\- - - - - - - - - - - - - - - - - - - -

1 2 3 4 5 6 7 8 9 10

Sum of first 10 natural numbers: 55

Q-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 method overloading.

PRACTICAL - 2

Source Code :

```
using System;

namespace AreaCalculator
{
   class Program
   {
      static void Main(string[] args)
       {
          ShapeCalculator calculator = new ShapeCalculator();

          double radiusCircle = 5.0;
          double circleArea = calculator.CalculateArea(radiusCircle);
          Console.WriteLine("Area of Circle with radius " + radiusCircle + ": " + circleArea);

          double lengthRectangle = 5.0;
          double widthRectangle = 10.0;
          double rectangleArea = calculator.CalculateArea(lengthRectangle, widthRectangle);
          Console.WriteLine("Area of Rectangle with length " + lengthRectangle + " and width " +
widthRectangle + ": " + rectangleArea);

          double sideSquare = 5.0;
          double squareArea = calculator.CalculateAreaSquare(sideSquare);
          Console.WriteLine("Area of Square with side " + sideSquare + ": " + squareArea);

          double side1Triangle = 5.0;
          double side2Triangle = 10.0;
```

PRACTICAL - 2

```
            double side3Triangle = 7.5;
            double triangleArea = calculator.CalculateAreaTriangle(side1Triangle, side2Triangle, side3Triangle);
            Console.WriteLine("Area of Triangle with sides " + side1Triangle + ", " + side2Triangle + ", and " + side3Triangle + ": " + triangleArea);
        }
    }
    class ShapeCalculator
    {
        public double CalculateArea(double radius)
        {
            return Math.PI * radius * radius;
        }
        public double CalculateArea(double length, double width)
        {
            return length * width;
        }
        public double CalculateAreaSquare(double side)
        {
            return side * side;
        }
        public double CalculateAreaTriangle(double side1, double side2, double side3)
        {
            double semiPerimeter = (side1 + side2 + side3) / 2.0;
            return Math.Sqrt(semiPerimeter * (semiPerimeter - side1) * (semiPerimeter - side2) * (semiPerimeter - side3));
        }
                                                                                    }
}
```

PRACTICAL - 2

OutPut :

Area of Circle with radius 5 : 78.5398163397448

Area of Rectangle with length 5 and width 10 : 50

Area of Square with side 5 : 25

Area of Triangle with sides 5, 10, and 7.5 : 18.1546094353473

Q-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.

Source Code :

```
using System;


class Program
{
   static void Main()
   {
      Console.Write("Enter the number of strings: ");
      int n = int.Parse(Console.ReadLine());


      string[] stringsArray = new string[n];
      int[] occurrenceArray = new int[n];


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

PRACTICAL - 2

```
        }

        int duplicateCount = 0;
        for (int i = 0; i < n; i++)
        {
            occurrenceArray[i] = 1; // Each string is at least present once (initial occurrence count)
            for (int j = i + 1; j < n; j++)
            {
                if (stringsArray[i] == stringsArray[j])
                {
                    duplicateCount++;
                    occurrenceArray[i]++;
                    stringsArray[j] = "";
                }
            }
        }
        Console.WriteLine($"Total number of duplicate strings:{duplicateCount}");

        for (int i = 0; i < n; i++)
        {
            if (occurrenceArray[i] > 1 && stringsArray[i] != "")
            {
                Console.WriteLine($"{stringsArray[i]} - Occurrence: {occurrenceArray[i]}");
            }
        }
    }
}
```

PRACTICAL - 2

OutPut :

Enter the number of strings: 3

Enter string 1: ABC Enter string

2: XYZ Enter string 3: ABC Total

number of duplicate strings:1

ABC - Occurrence: 2

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

Source Code :

```
using System;


class Program
{
   static void getMinMax(int[] arr, out int min, out int max)
   {
                                        min = int.MaxValue;
                                        max = int.MinValue;


      foreach (int num in arr)
       {
          if (num < min)
                    min = num;
                    if (num > max)

             max = num;
```

PRACTICAL - 2

```
    }
  }

  static void Main()
  {
    int[] numbers = { 7, 2, 9, 1, 5, 3 };

    Console.WriteLine("Input array:");
    foreach (int num in numbers)
    {
        Console.Write(num + " ");
    }

    int minValue, maxValue;
    getMinMax(numbers, out minValue, out maxValue);

    Console.WriteLine("\nMinimum value: " + minValue);
    Console.WriteLine("Maximum value: " + maxValue);

  }
}
```

OutPut :

Input array:

6 12 9 1 5 12

Minimum value: 1

Maximum value: 12

PRACTICAL - 2

Q-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.

Source Code :

```
using System;


class Program
{
   static void Main()
   {
      int[][] jaggedArray = new int[5][];
      jaggedArray[0] = new int[] { 7, 2,  9 };
      jaggedArray[1] = new int[] { 1,  5 };
      jaggedArray[2] = new int[] { 3, 8, 4, 6 };
      jaggedArray[3] = new int[] { 10, 12, 11 };
      jaggedArray[4] = new int[] { 15, 14, 13 };


      Console.WriteLine("Jagged array before sorting:");
      Console.WriteLine();
      DisplayJaggedArray(jaggedArray);
      for (int i = 0; i < jaggedArray.Length; i++)
      {
         Array.Sort(jaggedArray[i]);
      }
      Console.WriteLine("\nJagged array after sorting:");


      Console.WriteLine();
```

```
DisplayJaggedArray(jaggedArray);
 }
 static void DisplayJaggedArray(int[][] jaggedArray)
 {
 foreach (int[] subArray in jaggedArray)
 {
 foreach (int num in subArray)
 {
 Console.Write(num + " ");
 }
 Console.WriteLine();
 }
 }
}
```

Jagged array before sorting:

10 25 3

7 12 19 1

5

100 20 40 60 80

15 14

Jagged array after sorting:

3 10 25

1 7 12 19

5

20 40 60 80 100

14 15