Day 18(16-02-2022)
Assignment
By
Sudha Kumari Sugasani

Q1.What is the use of XML

XML-Extensible Markup Language
XML is used for universal data transfer mechanism to send across different platforms.

Q2.Write the points discussed about XML in the class

- XML stands for E**X**tensible Markup Language
- XML is a user defined tag.
- It is case sensitive.
- It can only have only one root tag.
- XML are two types

1. Tag based
2. Attribute based

- Tag based XML is taking more space to overcome that Attribute based XML is introduced.

Q3. Create a simple xml to illustrate:
a. Tag based xml with 10 products
b. Attribute based xml

a.Tag based XML with 10 Products

```
<Products>
 <Product>
  Id="1"
  Name="Mobile"
  Price="15000"
 </Product>
 <Product>
  Id="2"
  Name="NoteBook"
  Price="50"
 </Product>
 <Product>
  Id="3"
  Name="Pen"
  Price="15"
 </Product>
 <Product>
  Id="4"
  Name="Laptop"
  Price="60000"
 </Product>
 <Product>
  Id="5"
  Name="WaterBottle"
  Price="150"
 </Product>
 <Product>
```

```
 Id="6"
 Name="TravelBag"
 Price="5000"
</Product>
<Product>
 Id="7"
 Name="LEDTV"
 Price="45000"
</Product>
<Product>
 Id="8"
 Name="Alexa"
 Price="5000"
</Product>
<Product>
 Id="9"
 Name="Umbrella"
 Price="300"
</Product>
<Product>
 Id="10"
 Name="Watch"
 Price="2000"
</Product>
</Products>
```

Output:

```
<Products>
<Product> Id="1" Name="Mobile" Price="15000" </Product>
<Product> Id="2" Name="NoteBook" Price="50" </Product>
<Product> Id="3" Name="Pen" Price="15" </Product>
<Product> Id="4" Name="Laptop" Price="60000" </Product>
<Product> Id="5" Name="WaterBottle" Price="150" </Product>
<Product> Id="6" Name="TravelBag" Price="5000" </Product>
<Product> Id="7" Name="LEDTV" Price="45000" </Product>
<Product> Id="8" Name="Alexa" Price="5000" </Product>
<Product> Id="9" Name="Umbrella" Price="300" </Product>
<Product> Id="10" Name="Watch" Price="2000" </Product>
</Products>
```

```
▼<Products>
   <Product> Id="1" Name="Mobile" Price="15000" </Product>
   <Product> Id="2" Name="NoteBook" Price="50" </Product>
   <Product> Id="3" Name="Pen" Price="15" </Product>
   <Product> Id="4" Name="Laptop" Price="60000" </Product>
   <Product> Id="5" Name="WaterBottle" Price="150" </Product>
   <Product> Id="6" Name="TravelBag" Price="5000" </Product>
   <Product> Id="7" Name="LEDTV" Price="45000" </Product>
   <Product> Id="8" Name="Alexa" Price="5000" </Product>
   <Product> Id="9" Name="Umbrella" Price="300" </Product>
   <Product> Id="10" Name="Watch" Price="2000" </Product>
  </Products>
```

b.Attribute based XML with 10 products

```
<Products>
```

```
 <Product Id="1" Name="Mobile" Price="15000"/>
  <Product Id="2" Name="NoteBook" Price="50"/>
  <Product Id="3" Name="Pen" Price="15"/>
  <Product Id="4" Name="Laptop" Price="60000"/>
  <Product Id="5" Name="WaterBottle" Price="150"/>
  <Product Id="6" Name="TravelBag" Price="5000"/>
  <Product Id="7" Name="LEDTV" Price="45000"/>
  <Product Id="8" Name="Alexa" Price="5000"/>
  <Product Id="9" Name="Umbrella" Price="300"/>
  <Product Id="10" Name="Watch" Price="2000"/>
</Products>
```

Output:

```
<Products>
<Product Id="1" Name="Mobile" Price="15000"/>
<Product Id="2" Name="NoteBook" Price="50"/>
<Product Id="3" Name="Pen" Price="15"/>
<Product Id="4" Name="Laptop" Price="60000"/>
<Product Id="5" Name="WaterBottle" Price="150"/>
<Product Id="6" Name="TravelBag" Price="5000"/>
<Product Id="7" Name="LEDTV" Price="45000"/>
<Product Id="8" Name="Alexa" Price="5000"/>
<Product Id="9" Name="Umbrella" Price="300"/>
<Product Id="10" Name="Watch" Price="2000"/>
</Products>
```

```
▼<Products>
   <Product Id="1" Name="Mobile" Price="15000"/>
   <Product Id="2" Name="NoteBook" Price="50"/>
   <Product Id="3" Name="Pen" Price="15"/>
   <Product Id="4" Name="Laptop" Price="60000"/>
   <Product Id="5" Name="WaterBottle" Price="150"/>
   <Product Id="6" Name="TravelBag" Price="5000"/>
   <Product Id="7" Name="LEDTV" Price="45000"/>
   <Product Id="8" Name="Alexa" Price="5000"/>
   <Product Id="9" Name="Umbrella" Price="300"/>
   <Product Id="10" Name="Watch" Price="2000"/>
 </Products>
```

Q4.Convert the above XML to JSON and display the JSON data.

a.Converting tag based XML to JSON

```
[
   "Id=\"1\"\n   Name=\"Mobile\"\n   Price=\"15000\"",
   "Id=\"2\"\n   Name=\"NoteBook\"\n   Price=\"50\"",
   "Id=\"3\"\n   Name=\"Pen\"\n   Price=\"15\"",
   "Id=\"4\"\n   Name=\"Laptop\"\n   Price=\"60000\"",
   "Id=\"5\"\n   Name=\"WaterBottle\"\n   Price=\"150\"",
   "Id=\"6\"\n   Name=\"TravelBag\"\n   Price=\"5000\"",
   "Id=\"7\"\n   Name=\"LEDTV\"\n   Price=\"45000\"",
   "Id=\"8\"\n   Name=\"Alexa\"\n   Price=\"5000\"",
   "Id=\"9\"\n   Name=\"Umbrella\"\n   Price=\"300\"",
   "Id=\"10\"\n   Name=\"Watch\"\n   Price=\"2000\""
]
```

b.Converting attribute based XML to JSON


```
[
   {
      "@Id": "1",
      "@Name": "Mobile",
      "@Price": "15000"
   },
   {
      "@Id": "2",
      "@Name": "NoteBook",
      "@Price": "50"
   },
   {
      "@Id": "3",
      "@Name": "Pen",
      "@Price": "15"
   },
   {
      "@Id": "4",
      "@Name": "Laptop",
      "@Price": "60000"
   },
   {
      "@Id": "5",
      "@Name": "WaterBottle",
      "@Price": "150"
   },
   {
      "@Id": "6",
      "@Name": "TravelBag",
      "@Price": "5000"
   },
   {
      "@Id": "7",
      "@Name": "LEDTV",
      "@Price": "45000"
   },
   {
      "@Id": "8",
      "@Name": "Alexa",
      "@Price": "5000"
   },
   {
      "@Id": "9",
      "@Name": "Umbrella",
      "@Price": "300"
```

```
    },
    {
        "@Id": "10",
        "@Name": "Watch",
        "@Price": "2000"
    }
]
```

Q5.Research and write benefits of JSON over XML

- JSON takes less memory when compared to XML.
- JSON is easier to read than XML.
- JSON less tags than XML.

Q6. For the below requirement, create a layered architecture project with separate class library for Business logic.

create console application
create windows(or desktop) application

Business Requirement:

FIND FACTORIAL OF A NUMBER:

0 = 1

positive number (upto 7) = factorial answer

> 7 = -999 (as answer)

< 0 = -9999 (as answer)

put the screen shots of the output and project (solution explorer) screen shot

MathematicsLibrary Code:

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

namespace MathematicsLibrary
{
    /*************************************************************
     * Author:Sudha Kumari Sugasani
     * Purpose:Creating MathemematicsLibrary to reuse this in other
     *         libraries or applications.
     * ***********************************************************/
    public class Algebra
    {
        /// <summary>
        /// This method is used to get factorial of number
        /// </summary>
        /// <param name="n">int</param>
        /// <returns>Factorial value(int)</returns>
```

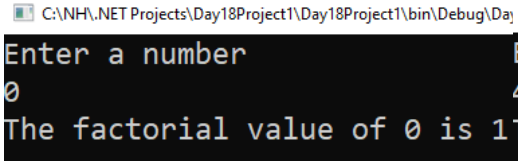
```
        public static int Factorial(int n)
        {
            if (n == 0)
                return 1;
            else if (n > 7)
                return -999;
            else if (n < 0)
                return -9999;
            else
            {
                int fact = 1;
                for(int i=1;i<=n;i++)
                {
                    fact = fact * i;
                }
                return fact;
            }
        }
    }
}
```

ConsoleApplication Code:

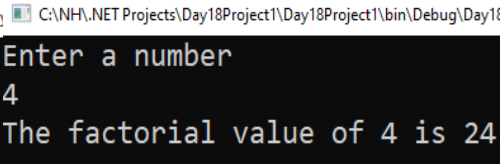
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MathematicsLibrary1;

namespace Day18Project2
{
    /*************************************************
     * Author:Sudha Kumari Sugasani
     * Purpose:Using MathematicsLibrary1 in consoleapp
     * ***********************************************/
    internal class Program
    {
        static void Main(string[] args)
        {
            int n;
            Console.WriteLine("Enter a number");
            n = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine($"Factorial of {n} is {Algebra.Factorial(n)}");
            Console.ReadLine();
        }
    }
}
```

Output:

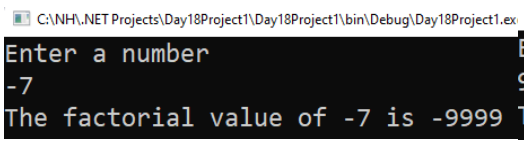
If Input is zero



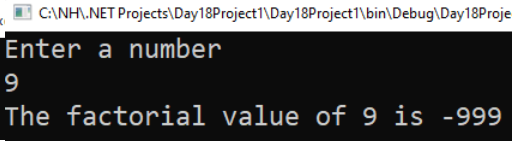
If Input is 1 to 7



If Input is <0



If Input is >7

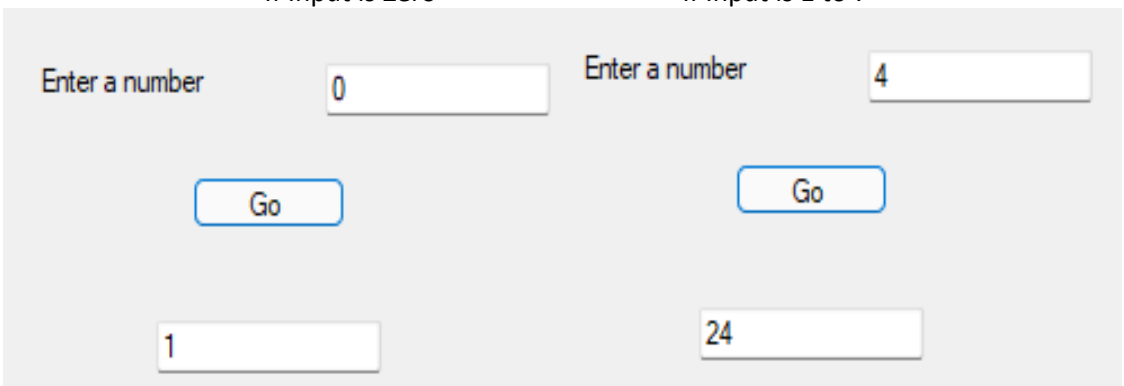


WebApplication Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MathematicsLibrary;

namespace MyApp
{
    /*****************************************************************
     * Author:Sudha Kumari Sugasani
     * Purpose:Creating a DesktopApplication using MathematicsLibrary
     *****************************************************************/
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            int n = Convert.ToInt32(textBox1.Text);
            int result = Algebra.Factorial(n);
            textBox2.Text = result.ToString();
            Console.ReadLine();

        }
    }
}
```

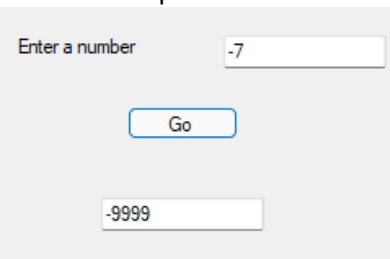
Output:

If Input is Zero

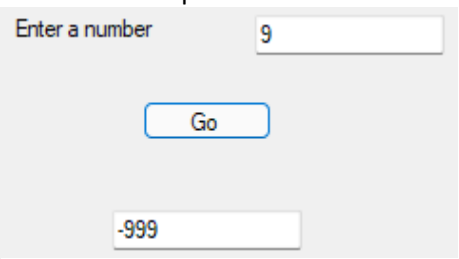


If Input is 1 to 7

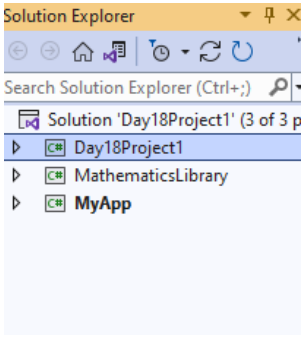
If Input is <0



If Input is >7



Project(Solution Explorer):



Q7.For the above method, Implement TDD
Write 4 test cases and put the code in word document.
Put the screen shot of all test cases failing
Make the test cases pass, put the screen shot.

MathematicsLibrary1 Code:

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

namespace MathematicsLibrary1
{

        /*************************************************************
         * Author:Sudha Kumari Sugasani
         * Purpose:Creating MathememeticsLibrary to reuse this in other
         *         Libraries or applications.
         * ***********************************************************/
        public class Algebra
        {
            /// <summary>
            /// This method is used to get factorial of number
            /// </summary>
            /// <param name="n">int</param>
            /// <returns>Factorial value(int)</returns>
            public static int Factorial(int n)
            {
            if (n == 0)
                return 1;
            else if (n > 7)
                return -999;
            else if (n < 0)
```

```
                return -9999;
            else
            {
                int fact = 1;
                for (int i = 1; i <= n; i++)
                {
                    fact = fact * i;
                }
                return fact;
            }

        }
    }
    }
```

MathematicsLibrary1Tests Code:

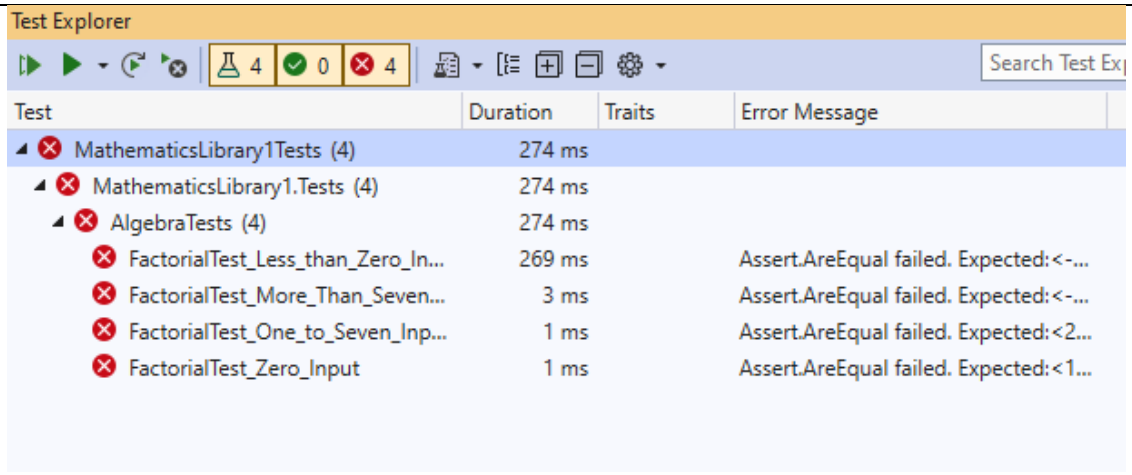
```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using MathematicsLibrary1;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace MathematicsLibrary1.Tests
{
    [TestClass()]
    public class AlgebraTests
    {
        /// <summary>
        /// This method is used to test if the input is zero
        /// </summary>
        [TestMethod()]
        public void FactorialTest_Zero_Input()
        {
            //Arrange
            int n = 0;
            int expected = 1;

            //Act
            int actual = Algebra.Factorial(n);

            //Assert
            Assert.AreEqual(expected, actual);

        }
        /// <summary>
        /// This method is used to test if the input is 1 to 7
        /// </summary>
        [TestMethod()]
        public void FactorialTest_One_to_Seven_Input()
        {
            //Arrange
            int n = 4;
            int expected = 24;
            //Act
            int actual = Algebra.Factorial(n);
            //Assert
            Assert.AreEqual(expected, actual);
        }
```

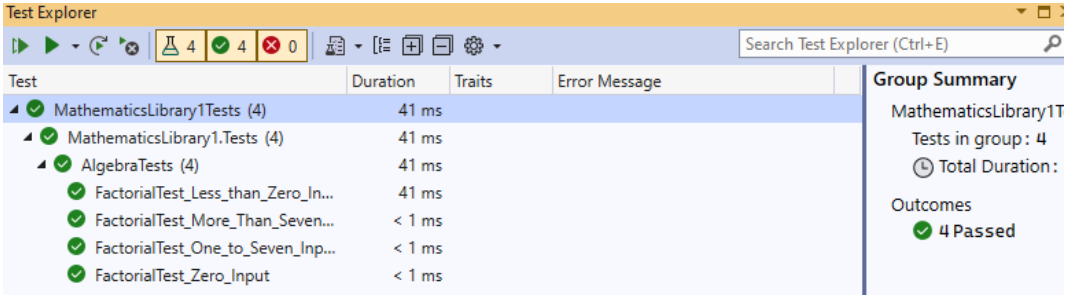
```
        /// <summary>
        /// This method id used to test if the input is >7
        /// </summary>
        [TestMethod()]
        public void FactorialTest_More_Than_Seven_Input()
        {
            //Arrange
            int n = 9;
            int expected = -999;
            //Act
            int actual = Algebra.Factorial(n);
            //Assert
            Assert.AreEqual(expected, actual);
        }
        /// <summary>
        /// This method is used to test if the input is <0
        /// </summary>
        [TestMethod()]
        public void FactorialTest_Less_than_Zero_Input()
        {
            //Arrange
            int n =-7;
            int expected = -9999;
            //Act
            int actual = Algebra.Factorial(n);
            //Assert
            Assert.AreEqual(expected, actual);
        }
    }
}
```

Console app Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MathematicsLibrary1;

namespace Day18Project2
{
    /*************************************************
     * Author:Sudha Kumari Sugasani
     * Purpose:Using MathematicsLibrary1 in consoleapp
     * ***********************************************/
    internal class Program
    {
        static void Main(string[] args)
        {
            int n;
            Console.WriteLine("Enter a number");
            n = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine($"Factorial of {n} is {Algebra.Factorial(n)}");
            Console.ReadLine();
        }
    }
}
```

Output:





Q8. Add one more method to check if the number is palindrome or not in the above Algebra class and write test case for the same.

MathematicsLibrary2 Code:

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

namespace MathematicsLibrary2
{
        /*************************************************************
         * Author:Sudha Kumari Sugasani
         * Purpose:Creating MathemematicsLibrary to reuse this in other
         *         Libraries or applications.
         * ***********************************************************/
        public class Algebra
        {

            /// <summary>
            /// This method is used to get factorial of number
            /// </summary>
            /// <param name="n">int</param>
            /// <returns>Factorial value(int)</returns>
            public static int Factorial(int n)
            {

            if (n == 0)
                return 1;
            else if (n > 7)
                return -999;
```

```
            else if (n < 0)
                return -9999;
            else
            {
                int fact = 1;
                for (int i = 1; i <= n; i++)
                {
                    fact = fact * i;
                }
                return fact;
            }


            }
            /// <summary>
            /// This method is used to check given number is palindrome or not
            /// </summary>
            /// <param name="input">int</param>
            public static bool isPalindromeorNot(int input)
            {
                int m, rem;
                int rev = 0;
                m = input;
               while(m>0)
               {
                    rem = m % 10;
                    m = m / 10;
                    rev = rev * 10 + rem;
               }
                if(input==rev)
                {
                    return true;

                }
                else
                {
                   return false;

                }


            }
        }
    }
```

MathematicsLibrary2Tests Code:

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using MathematicsLibrary2;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace MathematicsLibrary2.Tests
{
    [TestClass()]
    public class AlgebraTests
    {
```

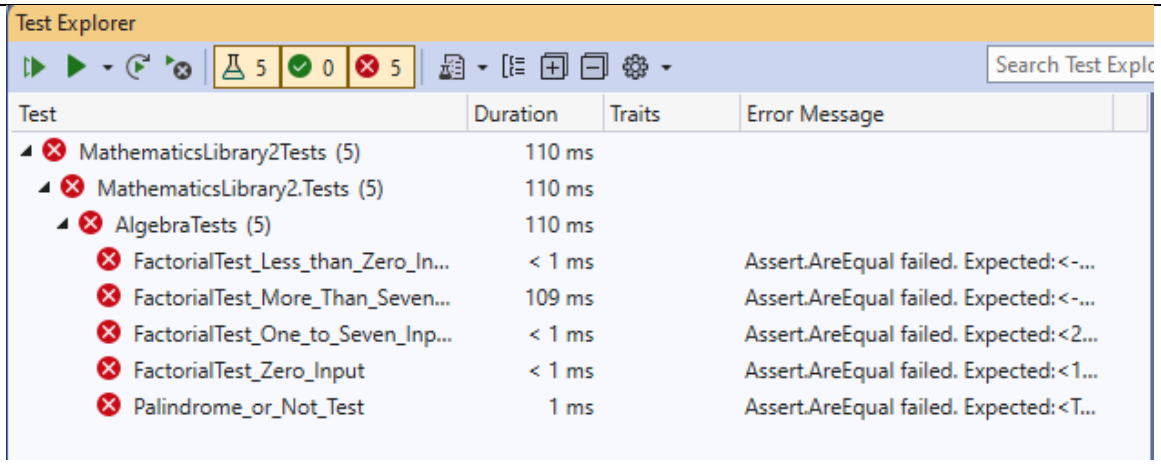
```
        /// <summary>
        /// This method is used to test if the input is zero
        /// </summary>
        [TestMethod()]
        public void FactorialTest_Zero_Input()
        {
            //Arrange
            int n = 0;
            int expected = 1;

            //Act
            int actual = Algebra.Factorial(n);

            //Assert
            Assert.AreEqual(expected, actual);

        }
        /// <summary>
        /// This method is used to test if the input is 1 to 7
        /// </summary>
        [TestMethod()]
        public void FactorialTest_One_to_Seven_Input()
        {
            //Arrange
            int n = 4;
            int expected = 24;
            //Act
            int actual = Algebra.Factorial(n);
            //Assert
            Assert.AreEqual(expected, actual);
        }
        /// <summary>
        /// This method id used to test if the input is >7
        /// </summary>
        [TestMethod()]
        public void FactorialTest_More_Than_Seven_Input()
        {
            //Arrange
            int n = 9;
            int expected = -999;
            //Act
            int actual = Algebra.Factorial(n);
            //Assert
            Assert.AreEqual(expected, actual);
        }
        /// <summary>
        /// This method is used to test if the input is <0
        /// </summary>
        [TestMethod()]
        public void FactorialTest_Less_than_Zero_Input()
        {
            //Arrange
            int n = -7;
            int expected = -9999;
            //Act
            int actual = Algebra.Factorial(n);
            //Assert
            Assert.AreEqual(expected, actual);
        }

        /// <summary>
        /// This method will check if the given number is palindrome or not
```

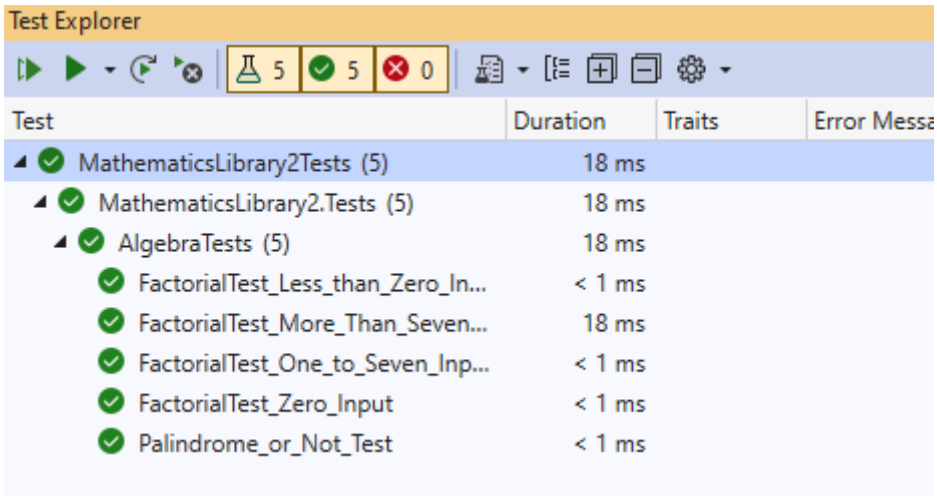
```
        /// </summary>
        [TestMethod()]
        public void Palindrome_or_Not_Test()
        {
            //Arrange
            int input = 121;
            bool expected = true;
            //Act
            bool actual = Algebra.isPalindromeorNot(input);
            //Assert
            Assert.AreEqual(expected,actual);
        }
    }
}
```

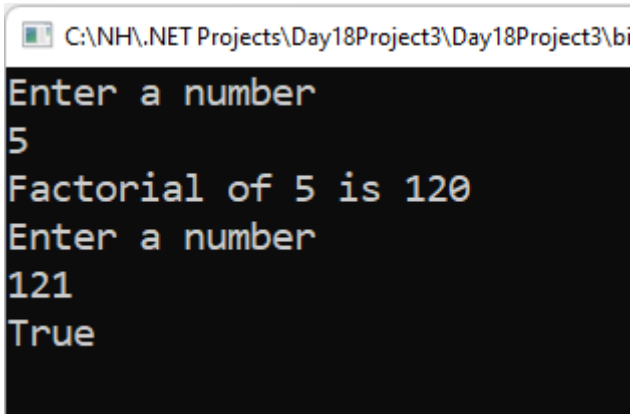
Console App Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MathematicsLibrary2;

namespace Day18Project3
{
    internal class Program
    {
        /*************************************************
          * Author:Sudha Kumari Sugasani
          * Purpose:Using MathematicsLibrary2 in consoleapp
          * ***********************************************/

        static void Main(string[] args)
        {
            //Algebra palindrome = new Algebra();
            int n;
            Console.WriteLine("Enter a number");
            n = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine($"Factorial of {n} is {Algebra.Factorial(n)}");
            int input;
            Console.WriteLine("Enter a number");
            input = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine(Algebra.isPalindromeorNot(input) );
            Console.ReadLine();

        }
    }
}
```

Output:


Test Explorer
5
0
5
Search Test Explo
Test
Duration
Traits
Error Message
MathematicsLibrary2Tests (5) 110 ms
MathematicsLibrary2.Tests (5) 110 ms
AlgebraTests (5) 110 ms
FactorialTest_Less_than_Zero_In... < 1 ms Assert.AreEqual failed. Expected:<-...
FactorialTest_More_Than_Seven... 109 ms Assert.AreEqual failed. Expected:<-...
FactorialTest_One_to_Seven_Inp... < 1 ms Assert.AreEqual failed. Expected:<2...
FactorialTest_Zero_Input < 1 ms Assert.AreEqual failed. Expected:<1...
Palindrome_or_Not_Test 1 ms Assert.AreEqual failed. Expected:<T...


Test Explorer
5
5
0
Test
Duration
Traits
Error Messa
MathematicsLibrary2Tests (5) 18 ms
MathematicsLibrary2.Tests (5) 18 ms
AlgebraTests (5) 18 ms
FactorialTest_Less_than_Zero_In... < 1 ms
FactorialTest_More_Than_Seven... 18 ms
FactorialTest_One_to_Seven_Inp... < 1 ms
FactorialTest_Zero_Input < 1 ms
Palindrome_or_Not_Test < 1 ms


C:\NH\.NET Projects\Day18Project3\Day18Project3\bi
Enter a number
5
Factorial of 5 is 120
Enter a number
121
True