.NET_WORKBOOK
NAME:P.Sweety
ID NO:2100030414 Lab 1: Tasks on C# Basics Concepts

PRE-LAB

1. What are the arithmetic Operators and Conditional statements in C#

Solution:

In C#, the arithmetic operators are:
1. Addition: `+`
2. Subtraction: `-`
3. Multiplication: `*`
4. Division: `/`
5. Modulus (Remainder): `%`

Conditional statements in C# include:
1. `if` statement
2. `else` statement
3. `else if` statement
4. `switch` statement

2. Answer the following

(i) What is Boxing and Un-Boxing with Example. **Solution:**

Boxing in C# is the process of converting a value type to the object or any interface type implemented by this value type. Example: int num = 10; object obj = num; Here, the integer value 10 is boxed into an object.

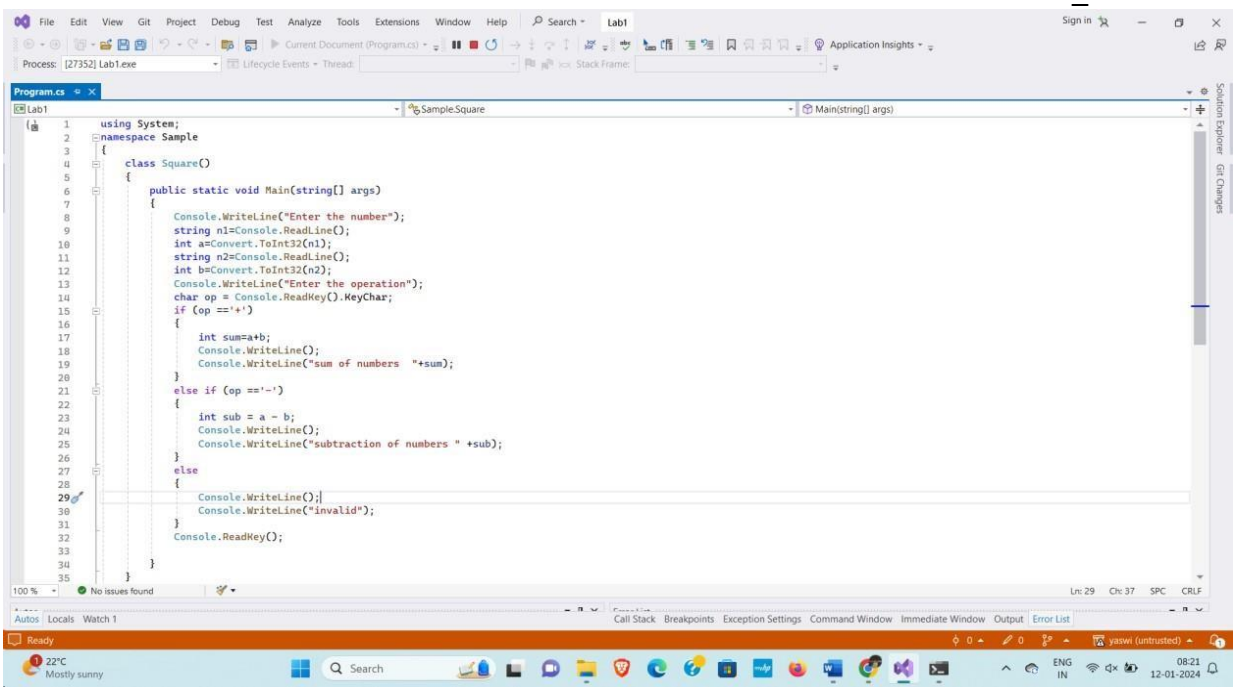
Unboxing, on the other hand, is the process of explicitly converting the previously boxed value type back to its original value type. Example: int num2 = (int)obj; Here, the boxed integer value is unboxed back to an integer variable.

IN-LAB:

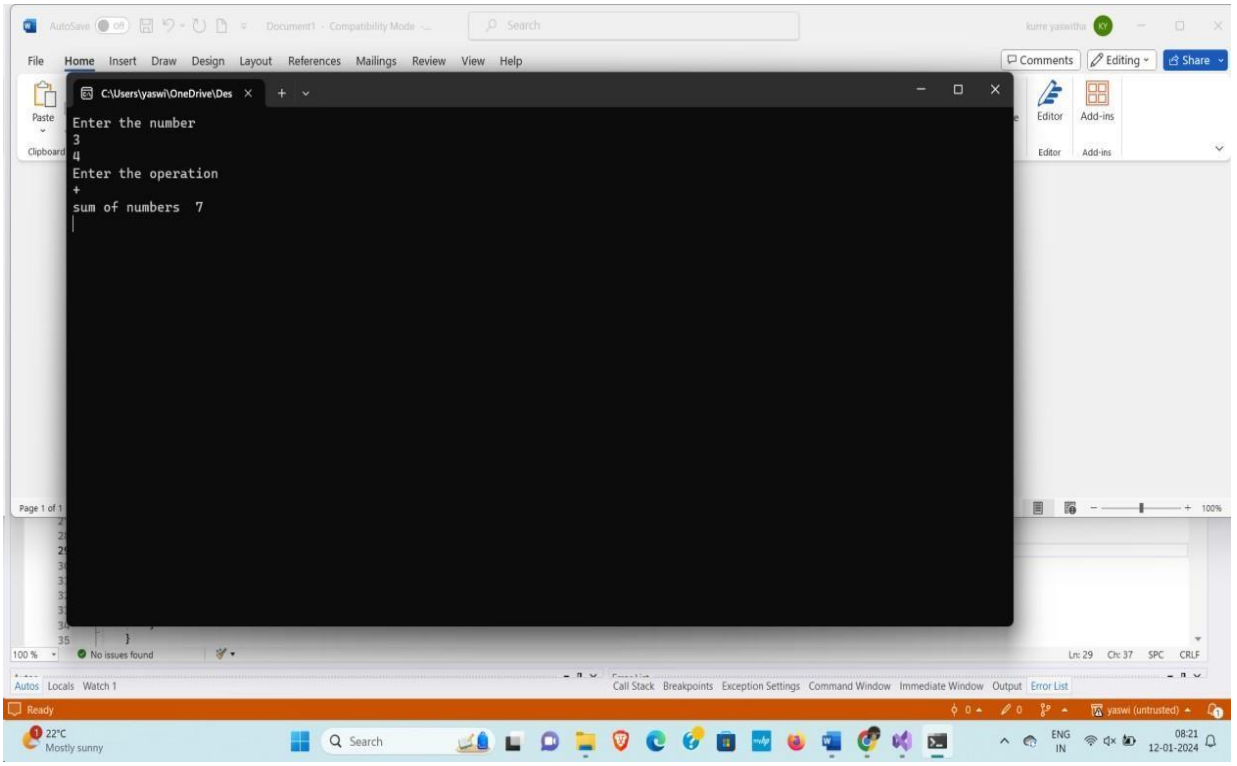
1. Write a C# code to implement the simple calculator?

TASK1: It's required to create a simple calculator with addition and subtraction operations for two integer number

Solution:



```
using System;
namespace Sample
{
    class Square()
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Enter the number");
            string n1=Console.ReadLine();
            int a=Convert.ToInt32(n1);
            string n2=Console.ReadLine();
            int b=Convert.ToInt32(n2);
            Console.WriteLine("Enter the operation");
            char op = Console.ReadKey().KeyChar;
            if (op =='+')
            {
                int sum=a+b;
                Console.WriteLine();
                Console.WriteLine("sum of numbers  "+sum);
            }
            else if (op =='-')
            {
                int sub = a - b;
                Console.WriteLine();
                Console.WriteLine("subtraction of numbers " +sub);
            }
            else
            {
                Console.WriteLine();
                Console.WriteLine("invalid");
            }
            Console.ReadKey();

        }
    }
```



```
Enter the number
3
4
Enter the operation
+
sum of numbers  7
```

2. Write a C# code to solve the TASK2 and TASK3.

TASK2: For a given integer *n* calculate the value which is equal to:

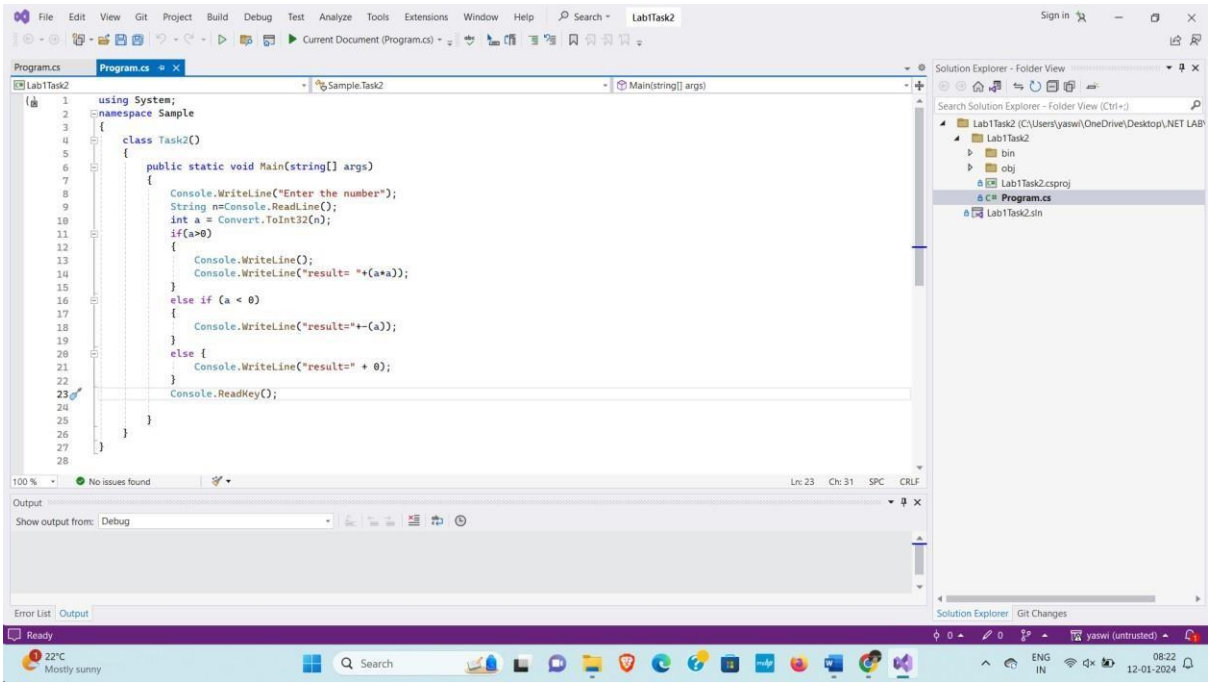
1. squared number, if its value is strictly positive; 2. modulus of a number, if its value is strictly negative;

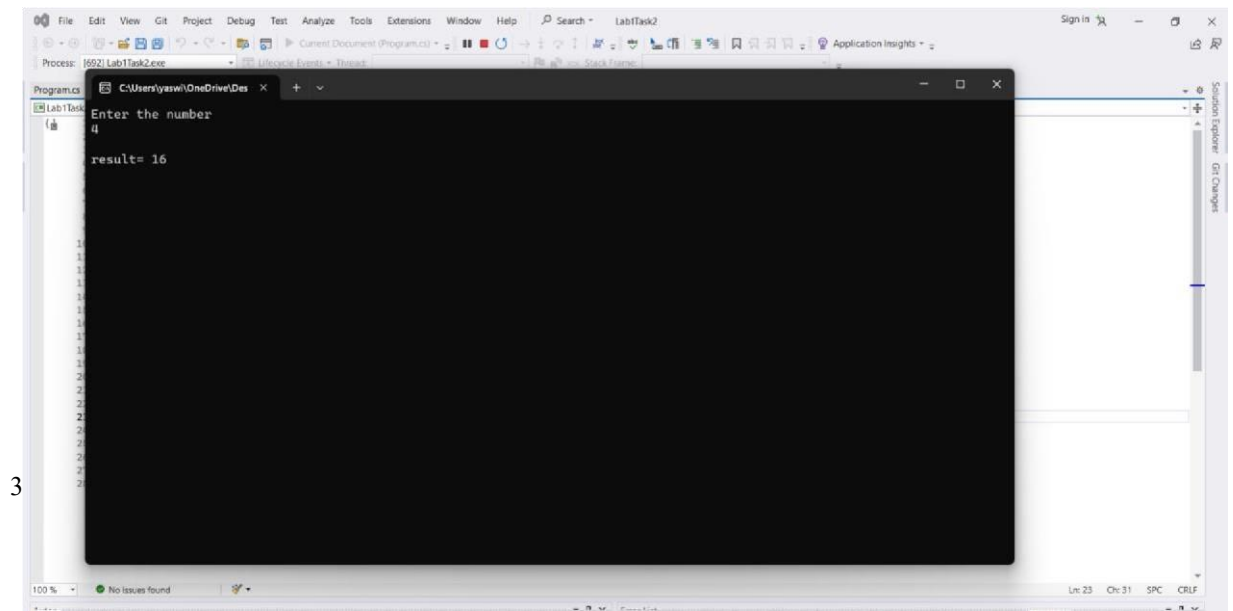
3. zero, if the integer n is zero. Example

```
n = 4      result =  16
n = -5     result =  5
n = 0      result = 0
```

TASK3: Find the maximum integer, that can be obtained by numbers of an arbitrary three-digit positive integer *n* permutation (100<=n<=999). Example `n = 165  result =  651`

Solution:

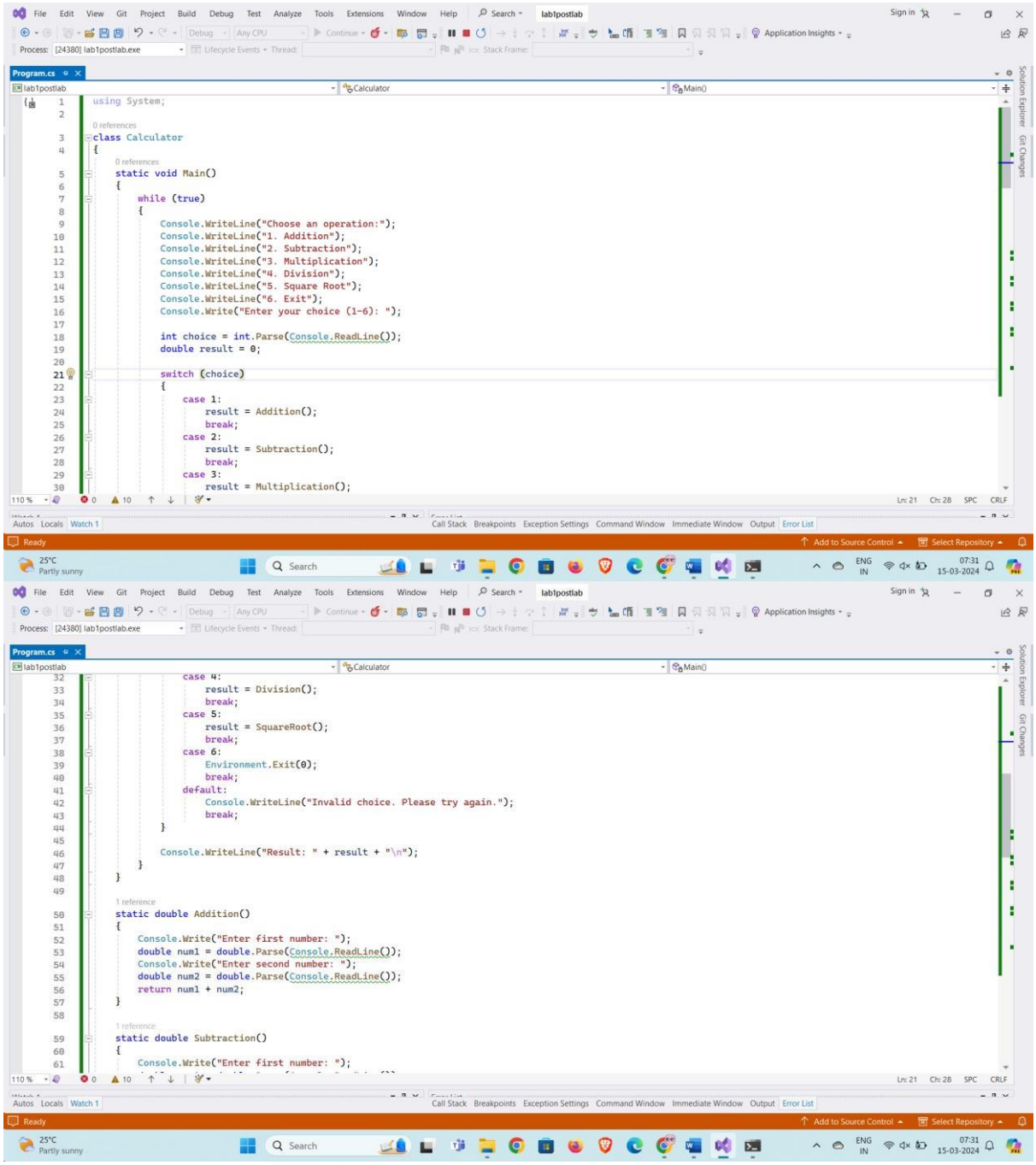




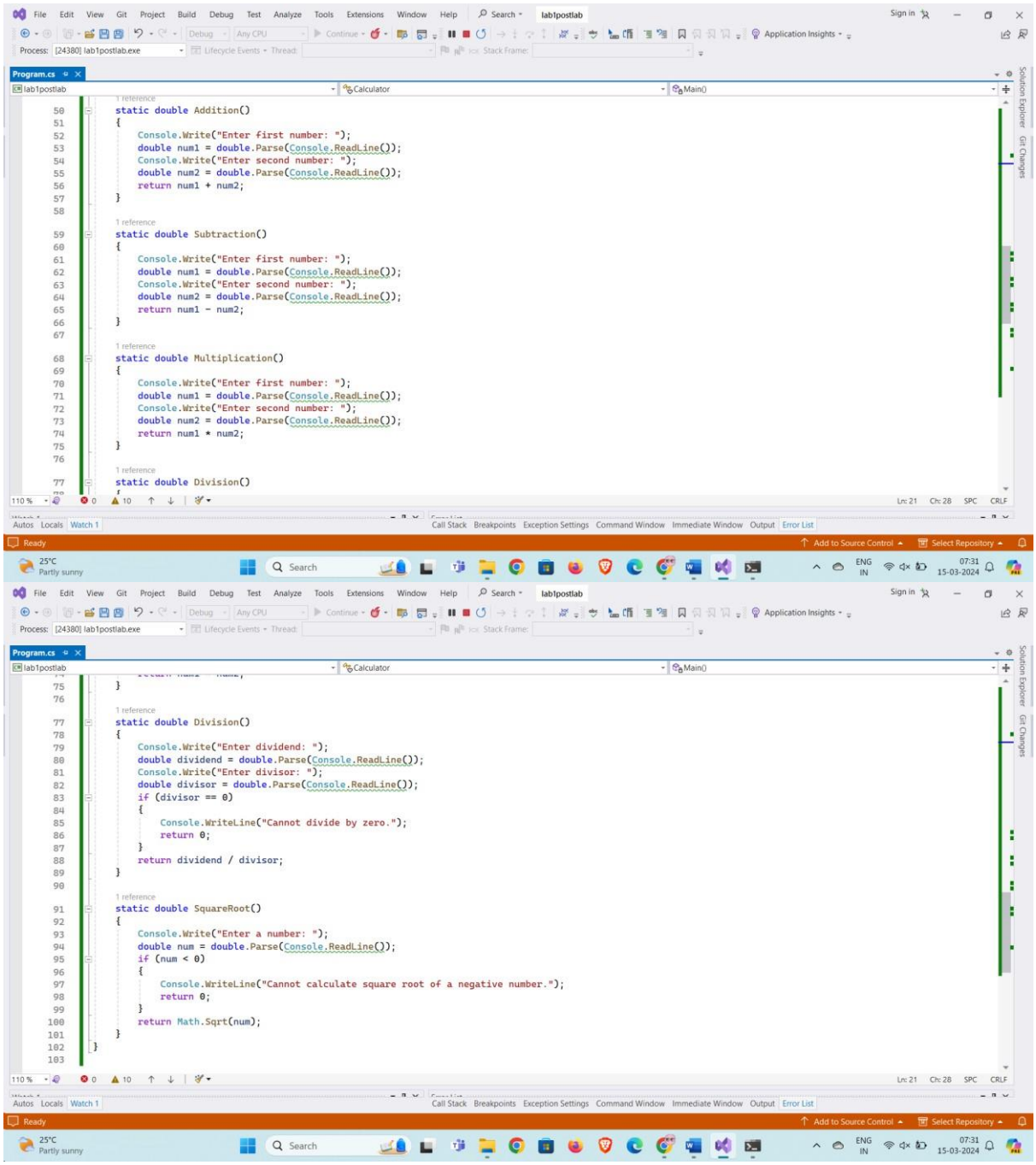
POST-LAB

1. Implement a proper calculator with all the functionalities like addition, subtraction, multiplication, division and square root.

Solution:



```
using System;

class Calculator
{
    static void Main()
    {
        while (true)
        {
            Console.WriteLine("Choose an operation:");
            Console.WriteLine("1. Addition");
            Console.WriteLine("2. Subtraction");
            Console.WriteLine("3. Multiplication");
            Console.WriteLine("4. Division");
            Console.WriteLine("5. Square Root");
            Console.WriteLine("6. Exit");
            Console.Write("Enter your choice (1-6): ");

            int choice = int.Parse(Console.ReadLine());
            double result = 0;

            switch (choice)
            {
                case 1:
                    result = Addition();
                    break;
                case 2:
                    result = Subtraction();
                    break;
                case 3:
                    result = Multiplication();
                    break;
                case 4:
                    result = Division();
                    break;
                case 5:
                    result = SquareRoot();
                    break;
                case 6:
                    Environment.Exit(0);
                    break;
                default:
                    Console.WriteLine("Invalid choice. Please try again.");
                    break;
            }

            Console.WriteLine("Result: " + result + "\n");
        }
    }

    static double Addition()
    {
        Console.Write("Enter first number: ");
        double num1 = double.Parse(Console.ReadLine());
        Console.Write("Enter second number: ");
        double num2 = double.Parse(Console.ReadLine());
        return num1 + num2;
    }

    static double Subtraction()
    {
        Console.Write("Enter first number: ");
```

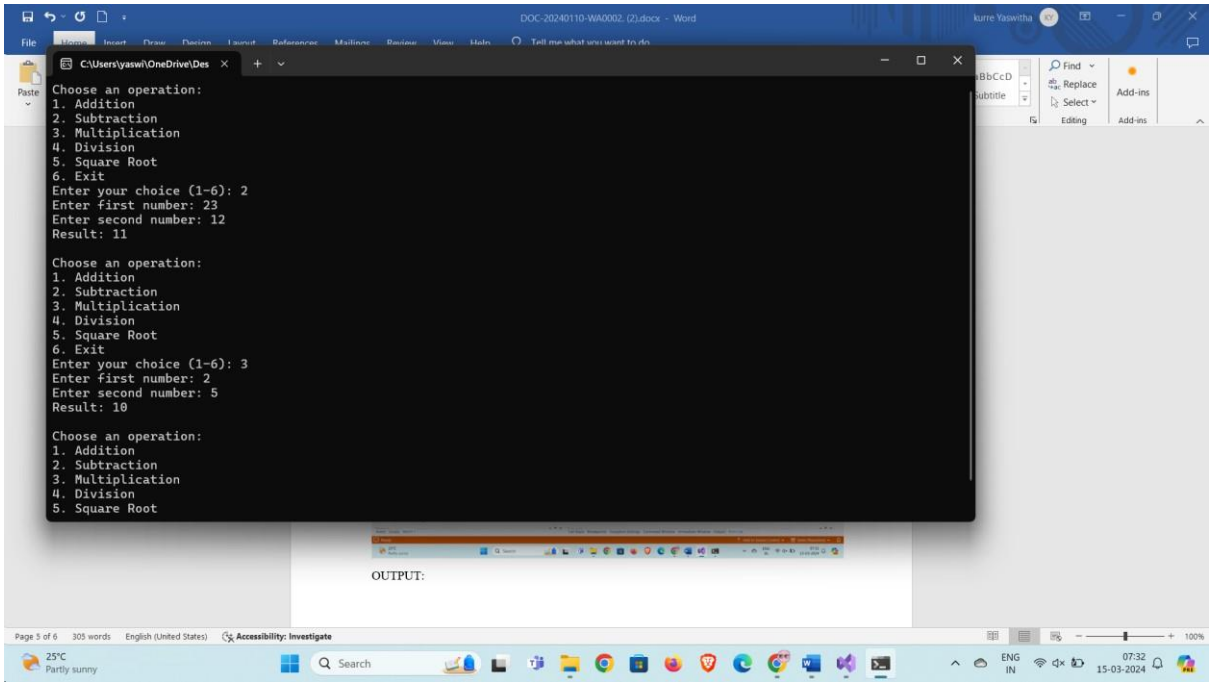


```
static double Addition()
{
    Console.Write("Enter first number: ");
    double num1 = double.Parse(Console.ReadLine());
    Console.Write("Enter second number: ");
    double num2 = double.Parse(Console.ReadLine());
    return num1 + num2;
}

static double Subtraction()
{
    Console.Write("Enter first number: ");
    double num1 = double.Parse(Console.ReadLine());
    Console.Write("Enter second number: ");
    double num2 = double.Parse(Console.ReadLine());
    return num1 - num2;
}

static double Multiplication()
{
    Console.Write("Enter first number: ");
    double num1 = double.Parse(Console.ReadLine());
    Console.Write("Enter second number: ");
    double num2 = double.Parse(Console.ReadLine());
    return num1 * num2;
}

static double Division()
```



```
}

static double Division()
{
    Console.Write("Enter dividend: ");
    double dividend = double.Parse(Console.ReadLine());
    Console.Write("Enter divisor: ");
    double divisor = double.Parse(Console.ReadLine());
    if (divisor == 0)
    {
        Console.WriteLine("Cannot divide by zero.");
        return 0;
    }
    return dividend / divisor;
}

static double SquareRoot()
{
    Console.Write("Enter a number: ");
    double num = double.Parse(Console.ReadLine());
    if (num < 0)
    {
        Console.WriteLine("Cannot calculate square root of a negative number.");
        return 0;
    }
    return Math.Sqrt(num);
}
}
```

OUTPUT:


C:\Users\yaswi\OneDrive\Des
Choose an operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Square Root
6. Exit
Enter your choice (1-6): 2
Enter first number: 23
Enter second number: 12
Result: 11
Choose an operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Square Root
6. Exit
Enter your choice (1-6): 3
Enter first number: 2
Enter second number: 5
Result: 10
Choose an operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Square Root
OUTPUT: