Jaya Krishna

ID:NO:210009062

Lab 2: Tasks on Looping, Arrays and Functions

PRE-LAB

1. Differentiate the iteration statement and goto statement

Solution:

Iteration statements (e.g., for, while, do-while) repeat a block of code based on a condition, while goto statement transfers control to a labeled statement, potentially disrupting the normal flow of execution.

2. Are we able to store dissimilar items in Array? if yes justify?

Solution:

Yes, we can store dissimilar items in an array if the array type is declared as an array of objects (`object[]`), as all types in C# are derived from the `object` type.

3.How many types of arrays in C#.net? and what are they?

Solution:

In C#.NET, there are three types of arrays:
1. Single-dimensional arrays
2. Multi-dimensional arrays
3. Jagged arrays

4.How can we initialize all the types of arrays in C#.Net?

Solution:

1. Single-dimensional array: int[] numbers = {1, 2, 3, 4, 5};
2. Multi-dimensional array: int[,] matrix = { {1, 2}, {3, 4}, {5, 6} };
3. Jagged array: int[][] jaggedArray = new int[3][] { new int[] {1, 2}, new int[] {3, 4, 5}, new int[] {6, 7, 8, 9} };

5.In how many ways can we define functions? Write the syntaxes.

Solution:

1.Instance Methods:

```
class MyClass {
   public void MyMethod() {
      // Method body
   }
}
```

2.Static Methods:

```
class MyClass {
   public static void MyStaticMethod() {
      // Method body
   }
}
```

3.Extension Methods:

```
public static class MyExtensions {
   public static void MyExtensionMethod(this string str) {
      // Method body
   }
}
```

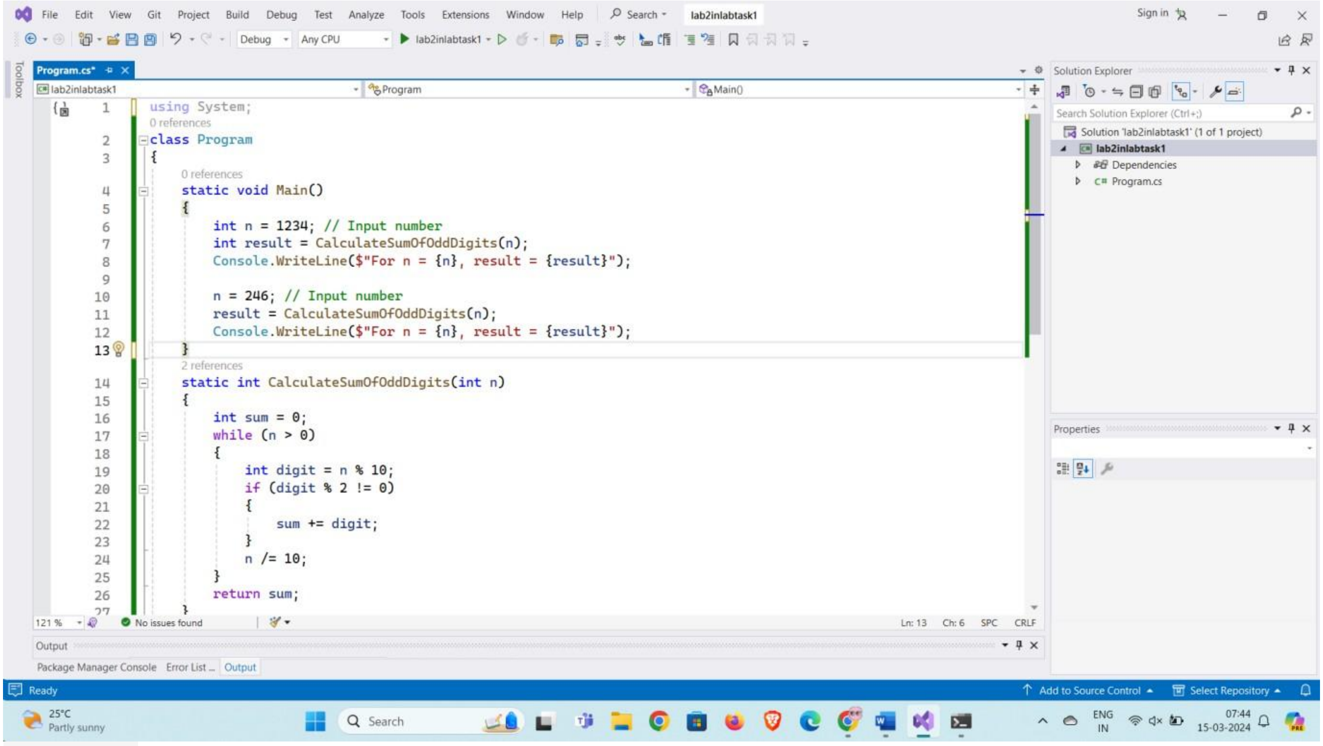
IN-LAB:

1. Write a C# code to implement the Tasks on Looping Statements?

TASK1: For a positive integer *n* calculate the *result* value, which is equal to the sum of the odd numbers in *n*

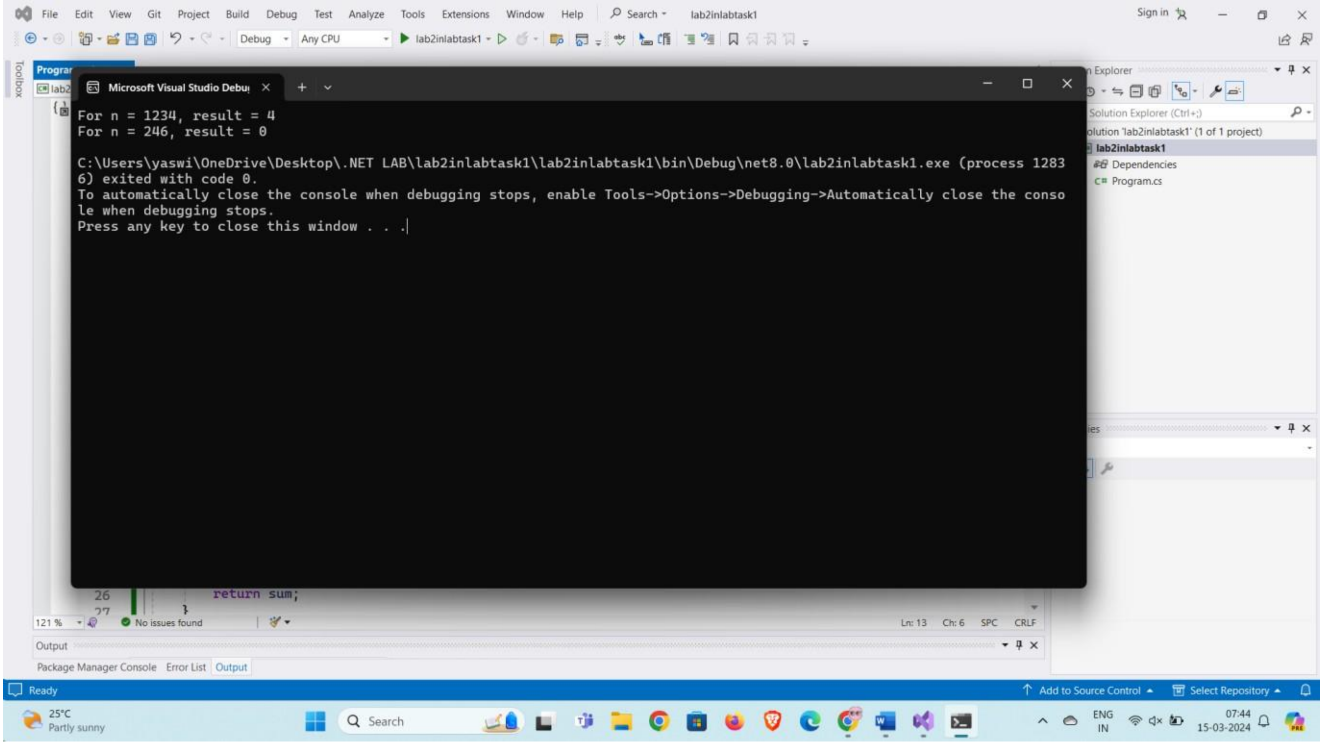
Example

```
n = 1234    result = 4  (1 + 3)
n = 246     result = 0
```



```
using System;
class Program
{
    static void Main()
    {
        int n = 1234; // Input number
        int result = CalculateSumOfOddDigits(n);
        Console.WriteLine($"For n = {n}, result = {result}");

        n = 246; // Input number
        result = CalculateSumOfOddDigits(n);
        Console.WriteLine($"For n = {n}, result = {result}");
    }
    static int CalculateSumOfOddDigits(int n)
    {
        int sum = 0;
        while (n > 0)
        {
            int digit = n % 10;
            if (digit % 2 != 0)
            {
                sum += digit;
            }
            n /= 10;
        }
        return sum;
    }
```

OUTPUT:



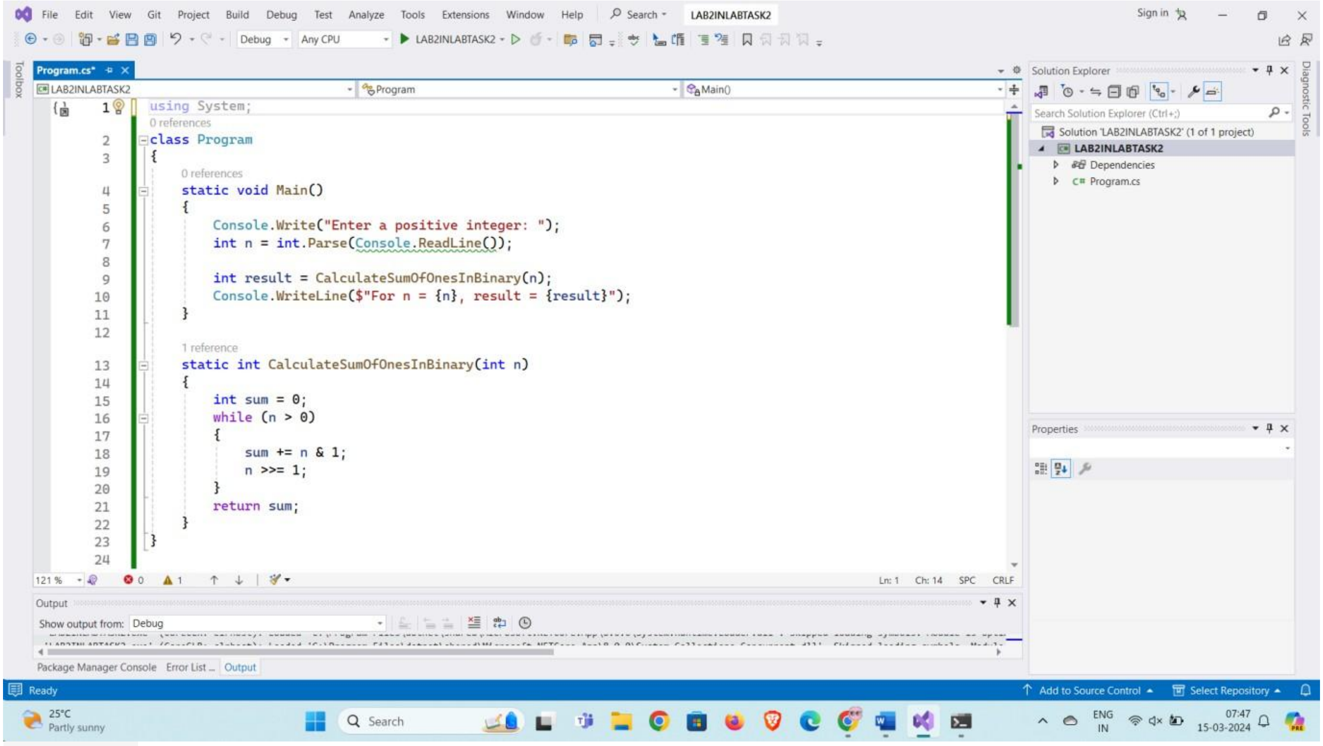
```
For n = 1234, result = 4
For n = 246, result = 0

C:\Users\yaswi\OneDrive\Desktop\.NET LAB\lab2inlabtask1\lab2inlabtask1\bin\Debug\net8.0\lab2inlabtask1.exe (process 1283
6) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

```
        return sum;
    }
```

TASK2: For a positive integer *n* calculate the result value, which is equal to the sum of the “1” in the binary representation of *n*.

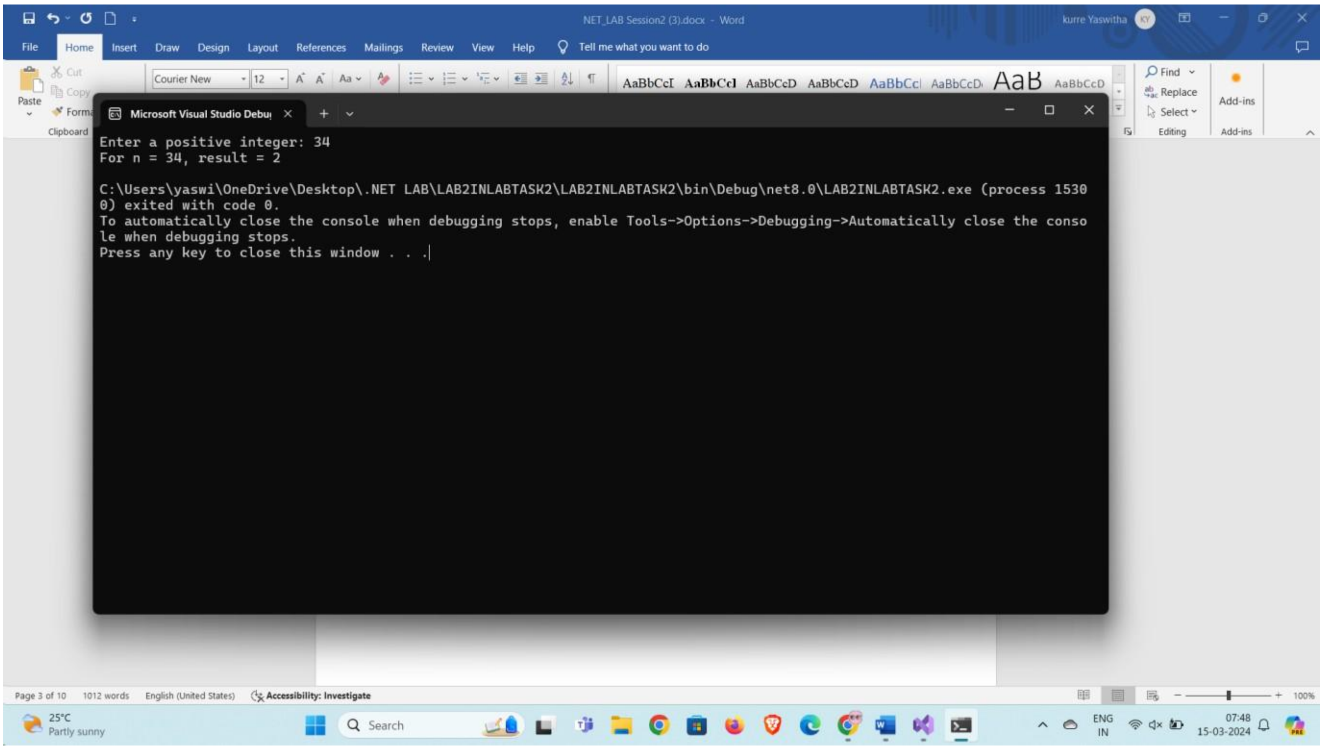
Example

```
n = 14(decimal) = 1110(binary)        result =  3
n = 128(decimal) = 1000 0000(binary) result  =  1
```



```csharp
using System;
class Program
{
    static void Main()
    {
        Console.Write("Enter a positive integer: ");
        int n = int.Parse(Console.ReadLine());

        int result = CalculateSumOfOnesInBinary(n);
        Console.WriteLine($"For n = {n}, result = {result}");
    }

    static int CalculateSumOfOnesInBinary(int n)
    {
        int sum = 0;
        while (n > 0)
        {
            sum += n & 1;
            n >>= 1;
        }
        return sum;
    }
}
```

OUTPUT:



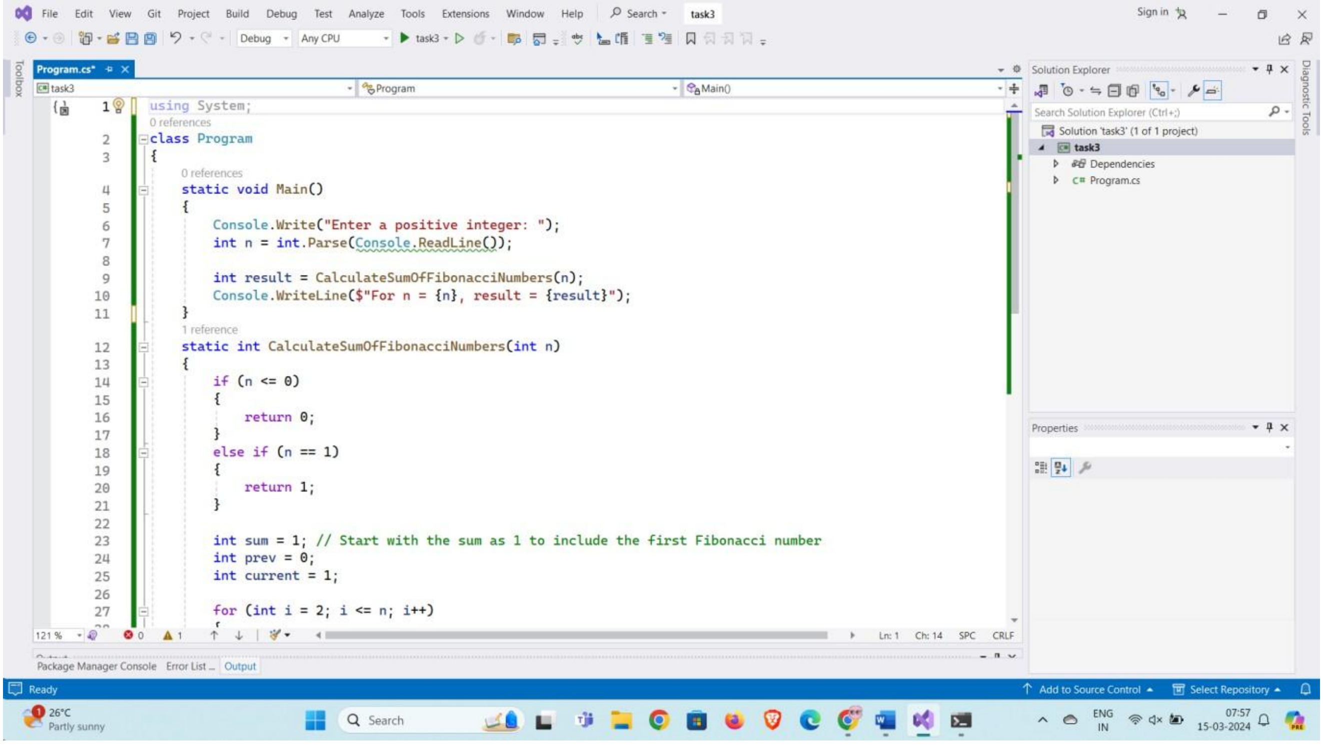
```
Enter a positive integer: 34
For n = 34, result = 2

C:\Users\yaswi\OneDrive\Desktop\.NET LAB\LAB2INLABTASK2\LAB2INLABTASK2\bin\Debug\net8.0\LAB2INLABTASK2.exe (process 1530
0) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

TASK3: For a positive integer *n*, calculate the result value equal to the sum of the first *n* Fibonacci numbers Note: Fibonacci numbers are a series of numbers in which each next number is equal to the sum of the two preceding ones: 0, 1, 1, 2, 3, 5, 8, 13... (F0=0, F1=F2=1, then F(n)=F(n-1)+F(n-2) for n>2)

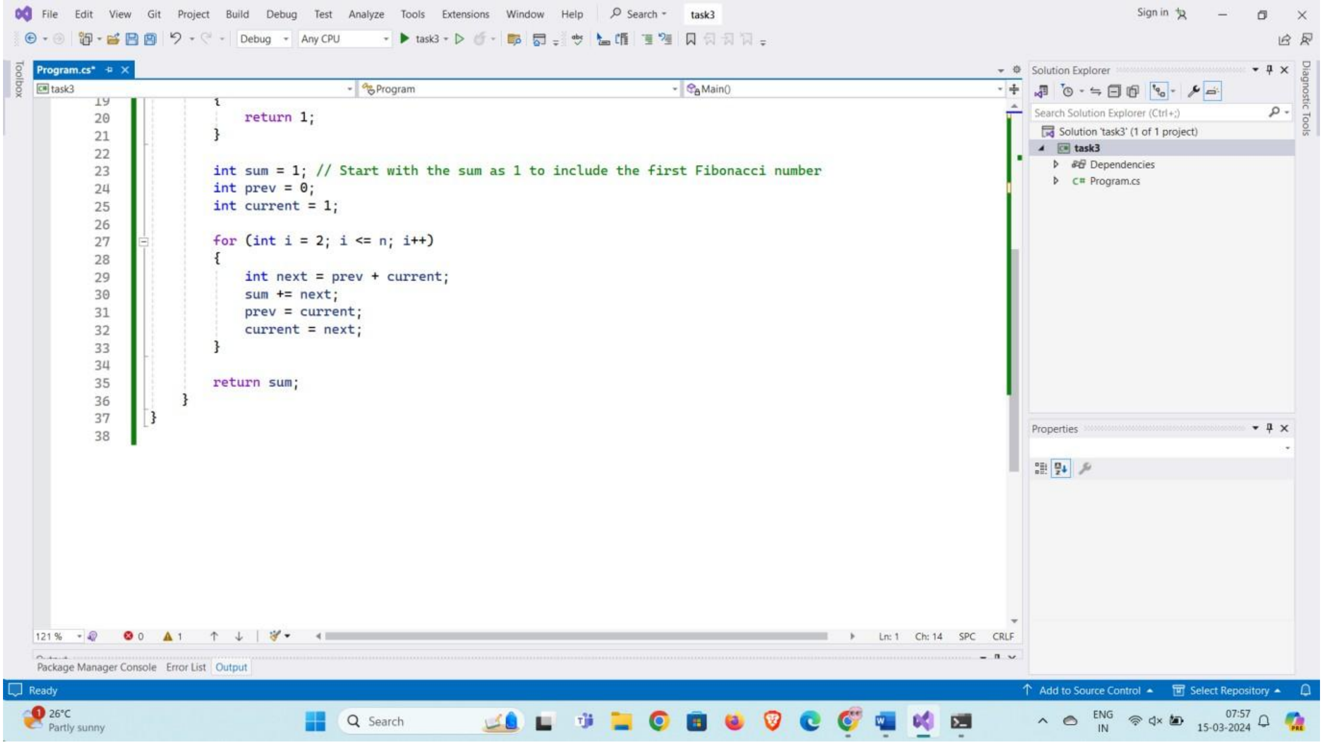
Example

```
n = 8     result = 33
n = 11    result = 143
```

Solution:

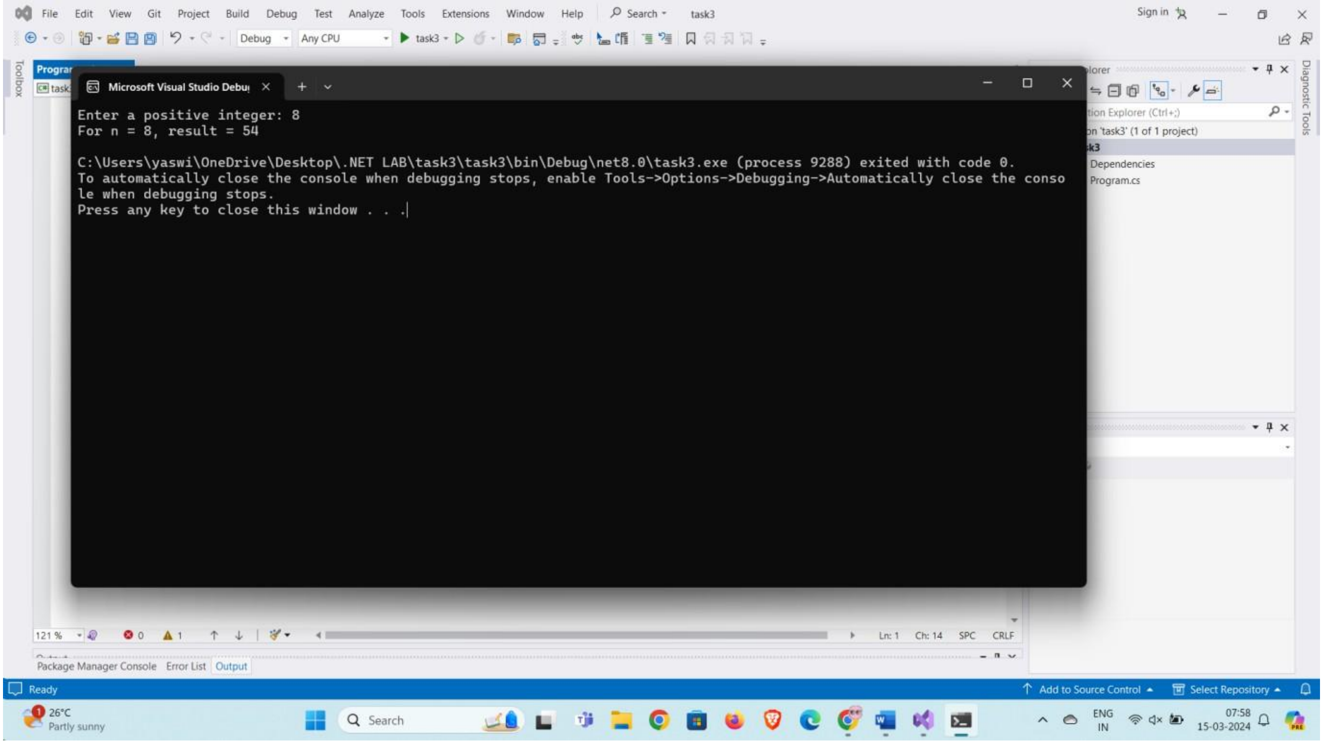


```
using System;
0 references
class Program
{
    0 references
    static void Main()
    {
        Console.Write("Enter a positive integer: ");
        int n = int.Parse(Console.ReadLine());

        int result = CalculateSumOfFibonacciNumbers(n);
        Console.WriteLine($"For n = {n}, result = {result}");
    }
    1 reference
    static int CalculateSumOfFibonacciNumbers(int n)
    {
        if (n <= 0)
        {
            return 0;
        }
        else if (n == 1)
        {
            return 1;
        }

        int sum = 1; // Start with the sum as 1 to include the first Fibonacci number
        int prev = 0;
        int current = 1;

        for (int i = 2; i <= n; i++)
```



```
            {
                return 1;
            }

            int sum = 1; // Start with the sum as 1 to include the first Fibonacci number
            int prev = 0;
            int current = 1;

            for (int i = 2; i <= n; i++)
            {
                int next = prev + current;
                sum += next;
                prev = current;
                current = next;
            }

            return sum;
        }
    }
```

OUTPUT:



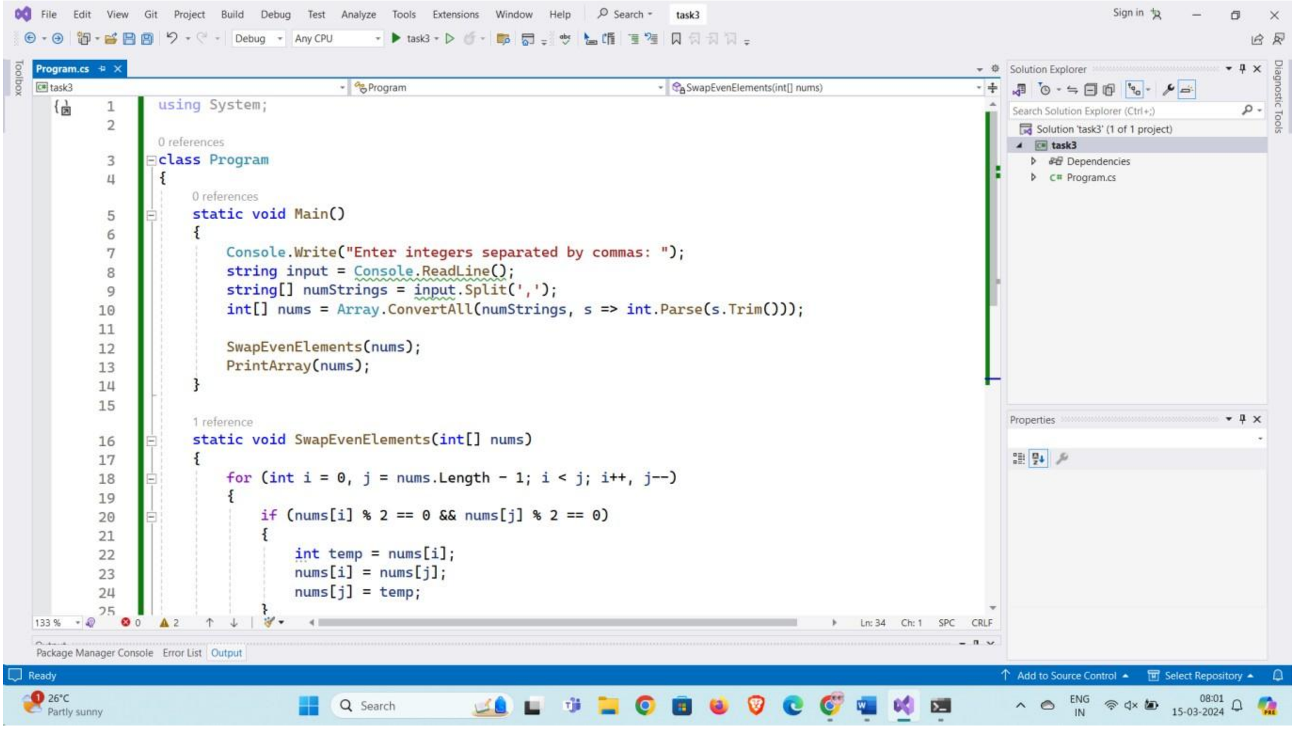
```
Enter a positive integer: 8
For n = 8, result = 54

C:\Users\yaswi\OneDrive\Desktop\.NET LAB\task3\task3\bin\Debug\net8.0\task3.exe (process 9288) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

2. Write a C# code to implement the Tasks on Arrays?

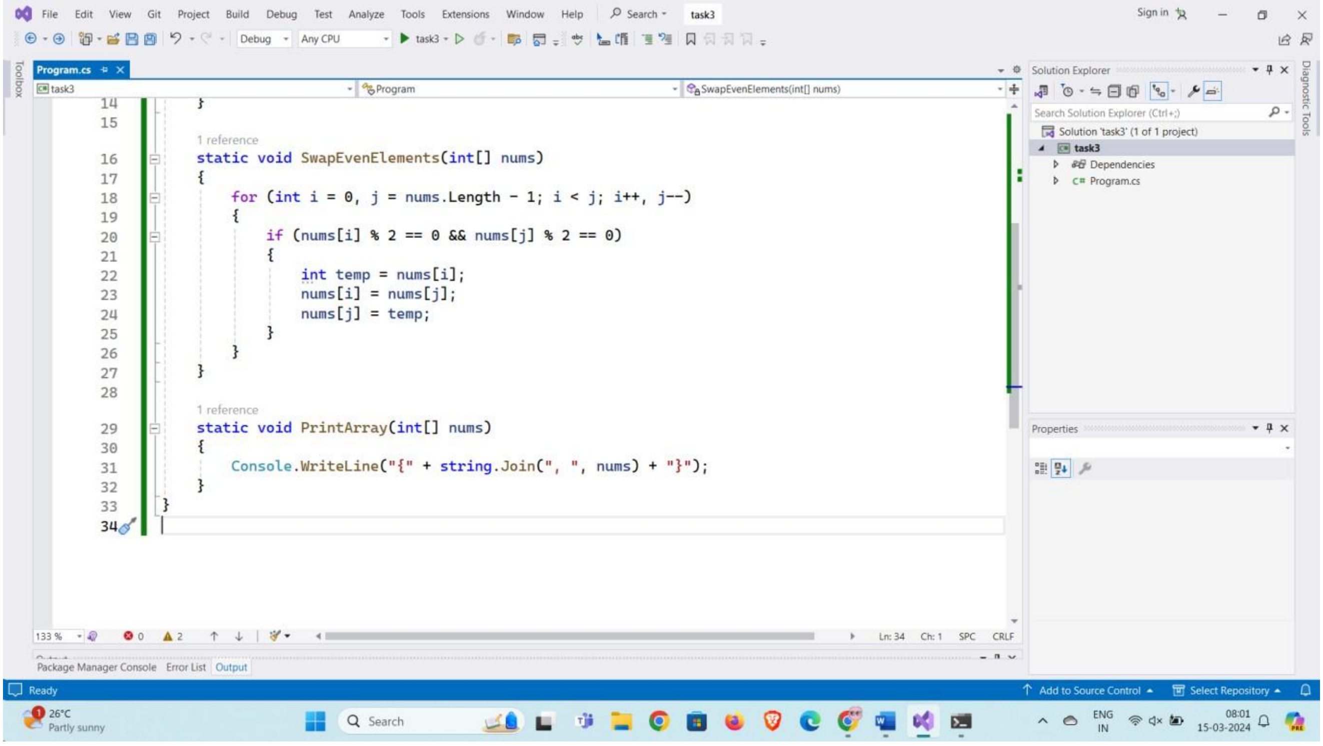
TASK 1: In a given array of integers ***nums*** swap values of the first and the last array elements, the second and the penultimate etc., if the two exchanged values are even

Example

```
{ 10 , 5, 3, 4}                          =>  {4, 5, 3, 10}
{100, 2, 3, 4, 5}                       => {100, 4, 3, 2, 5}
{100, 2, 3, 45, 33, 8, 4, 54} => {54, 4, 3, 45, 33, 8, 2, 100}
```

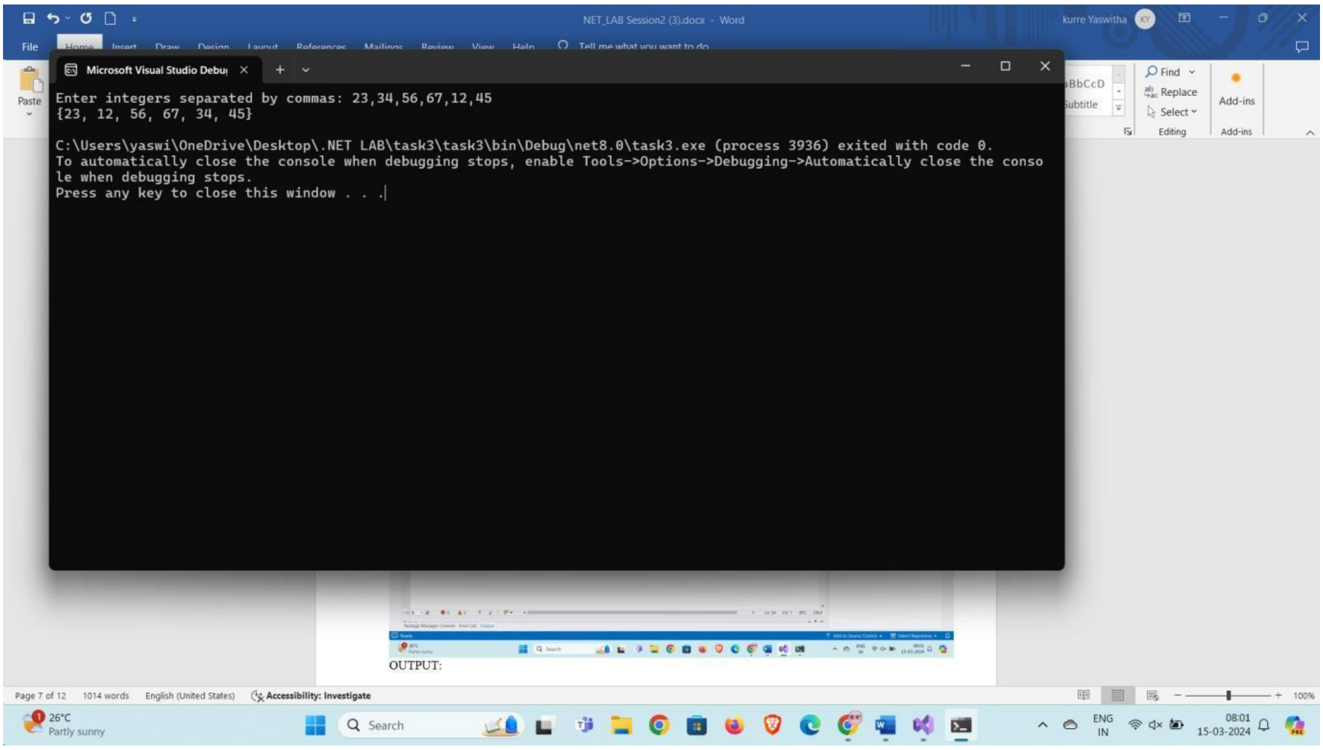


```
using System;

class Program
{
    static void Main()
    {
        Console.Write("Enter integers separated by commas: ");
        string input = Console.ReadLine();
        string[] numStrings = input.Split(',');
        int[] nums = Array.ConvertAll(numStrings, s => int.Parse(s.Trim()));

        SwapEvenElements(nums);
        PrintArray(nums);
    }

    static void SwapEvenElements(int[] nums)
    {
        for (int i = 0, j = nums.Length - 1; i < j; i++, j--)
        {
            if (nums[i] % 2 == 0 && nums[j] % 2 == 0)
            {
                int temp = nums[i];
                nums[i] = nums[j];
                nums[j] = temp;
            }
```



```
        }

        static void SwapEvenElements(int[] nums)
        {
            for (int i = 0, j = nums.Length - 1; i < j; i++, j--)
            {
                if (nums[i] % 2 == 0 && nums[j] % 2 == 0)
                {
                    int temp = nums[i];
                    nums[i] = nums[j];
                    nums[j] = temp;
                }
            }
        }

        static void PrintArray(int[] nums)
        {
            Console.WriteLine("{" + string.Join(", ", nums) + "}");
        }
    }
```

OUTPUT:



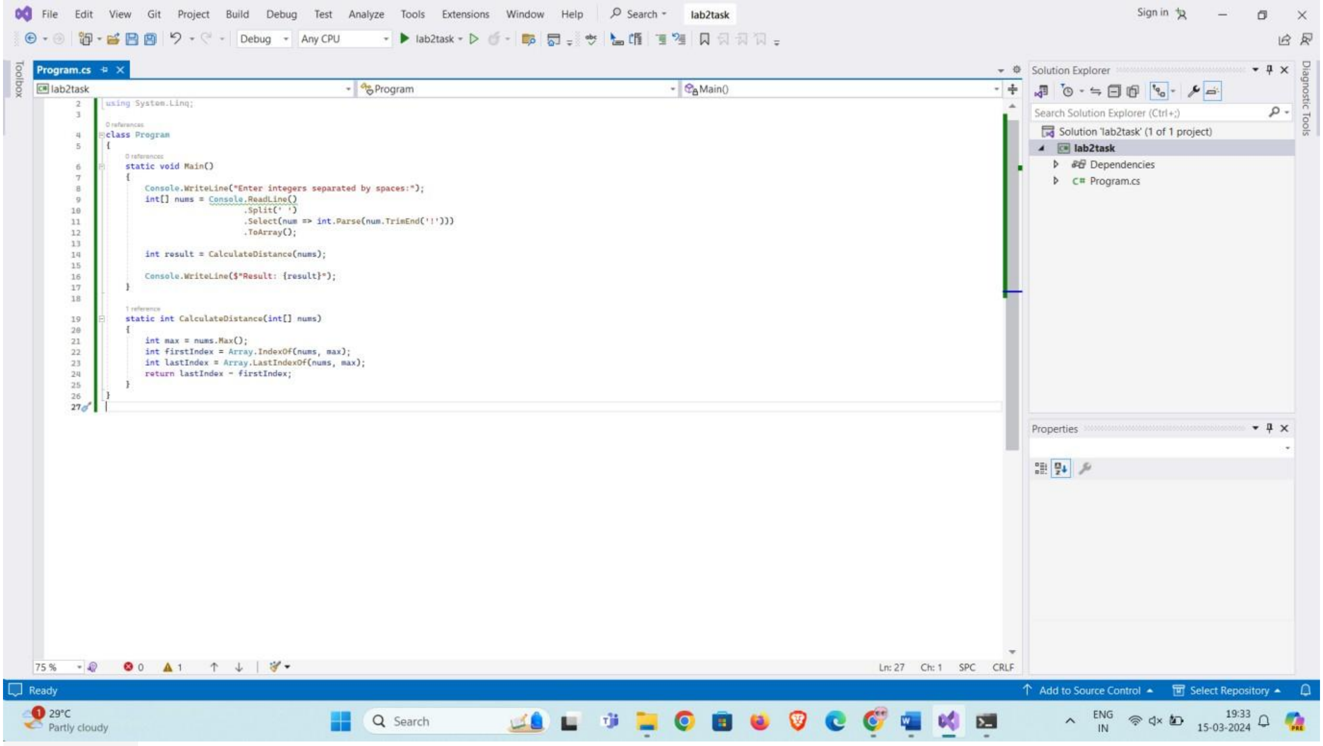
```
Enter integers separated by commas: 23,34,56,67,12,45
{23, 12, 56, 67, 34, 45}

C:\Users\yaswi\OneDrive\Desktop\.NET LAB\task3\task3\bin\Debug\net8.0\task3.exe (process 3936) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

TASK 2:

In a given array of integers ***nums*** calculate integer ***result*** value, that is equal to the distance between the first and the last entry of the maximum value in the array.

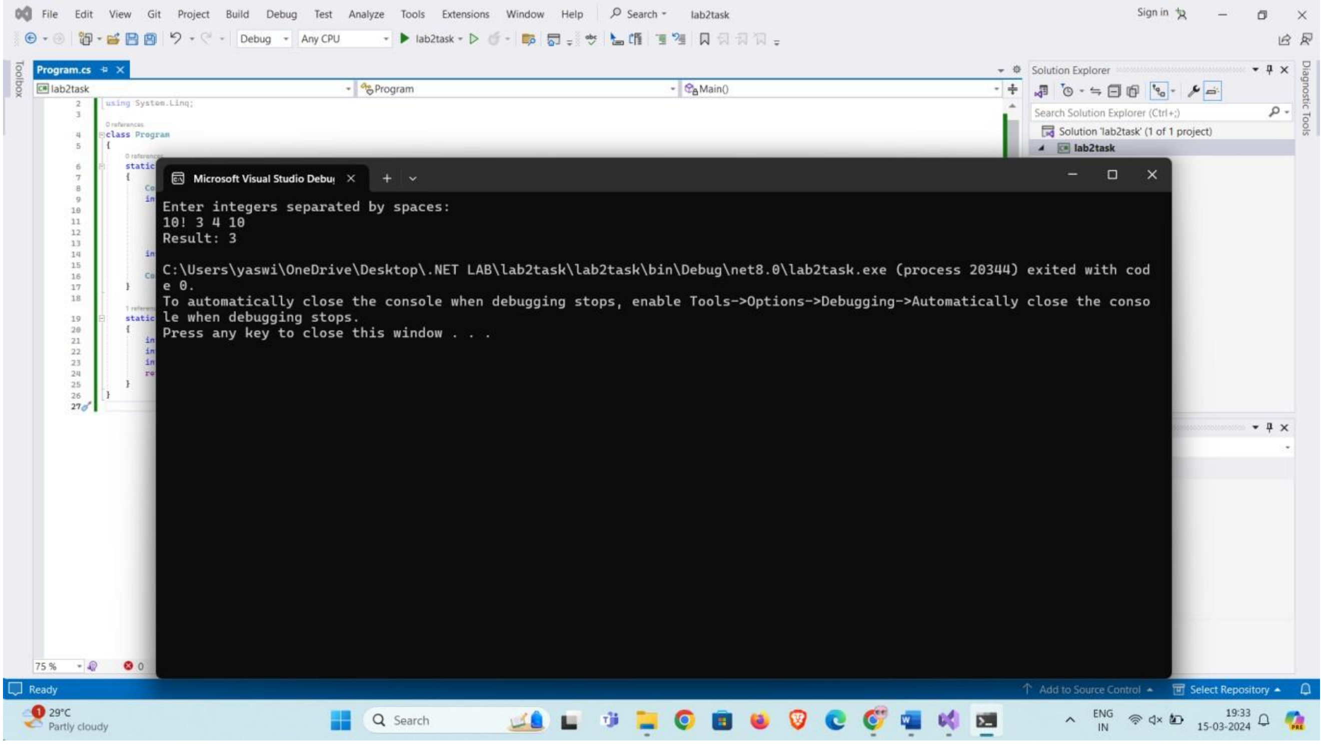
Example

```
{4, 100!, 3, 4}                result = 0
{5, 50!, 50!, 4, 5}          result = 1
{5, 350!, 350, 4, 350!} result = 3
{10!, 10, 10, 10, 10!}    result = 4
```



```
using System.Linq;

class Program
{
    static void Main()
    {
        Console.WriteLine("Enter integers separated by spaces:");
        int[] nums = Console.ReadLine()
                            .Split(' ')
                            .Select(num => int.Parse(num.TrimEnd('!')))
                            .ToArray();

        int result = CalculateDistance(nums);

        Console.WriteLine($"Result: {result}");
    }

    static int CalculateDistance(int[] nums)
    {
        int max = nums.Max();
        int firstIndex = Array.IndexOf(nums, max);
        int lastIndex = Array.LastIndexOf(nums, max);
        return lastIndex - firstIndex;
    }
}
```

OUTPUT:



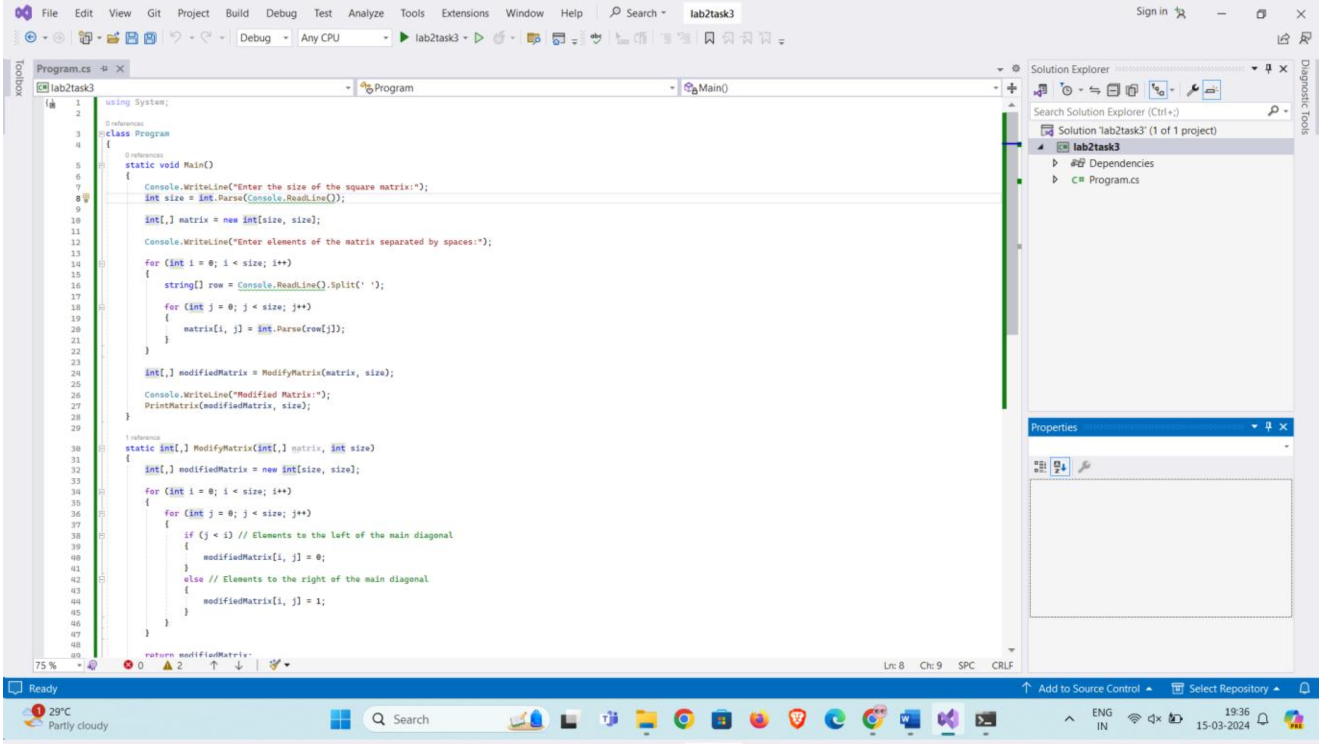
```
Enter integers separated by spaces:
10! 3 4 10
Result: 3

C:\Users\yaswi\OneDrive\Desktop\.NET LAB\lab2task\lab2task\bin\Debug\net8.0\lab2task.exe (process 20344) exited with cod
e 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

TASK 3: In a predetermined two-dimensional integer array (square matrix) ***matrix*** insert 0 into elements to the left side of the main diagonal, and 1 into elements to the right side of the diagonal.

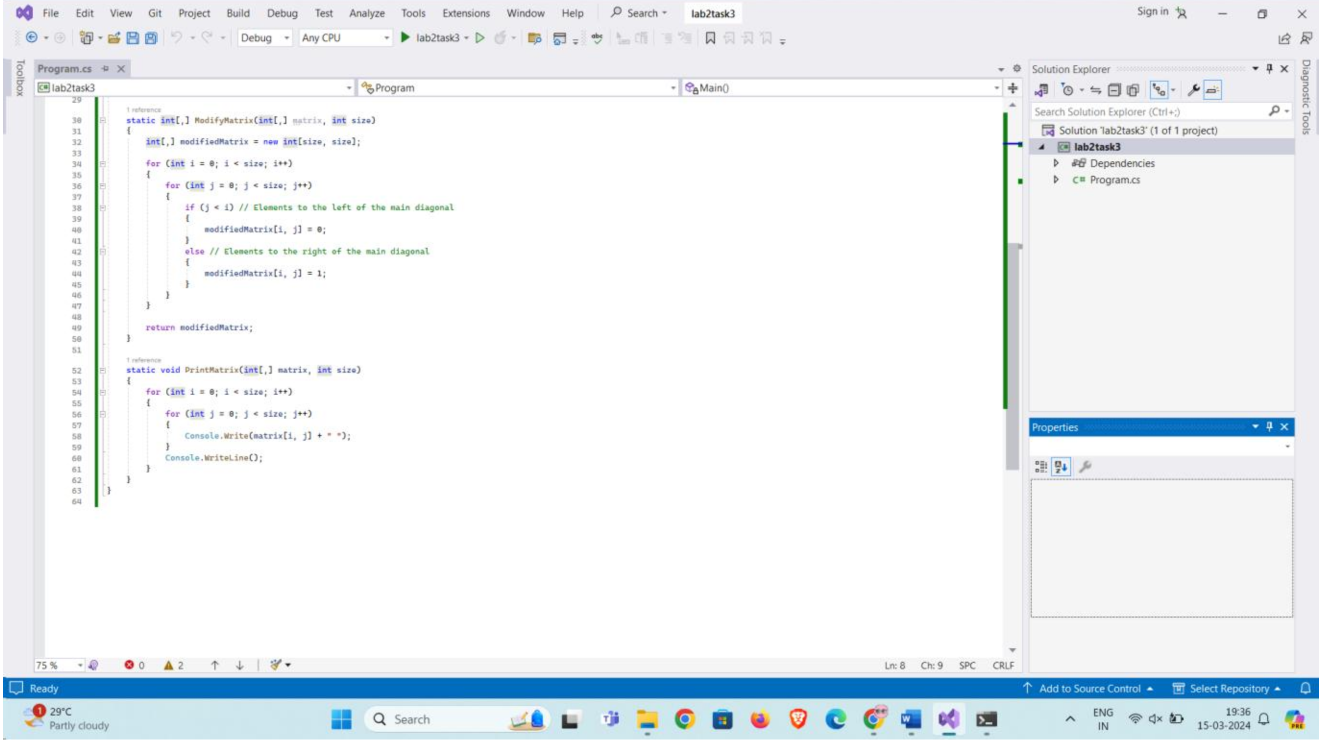
Example

```
{{2,  4,  3,  3},              {{2,  1,  1,  1},
 {5,  7,  8,  5},     =>       {0,  7,  1,  1},
 {2,  4,  3,  3},                 {0,  0,  3,  1},
{5, 7, 8, 5}}                {0, 0, 0, 5}}
```

Solution:



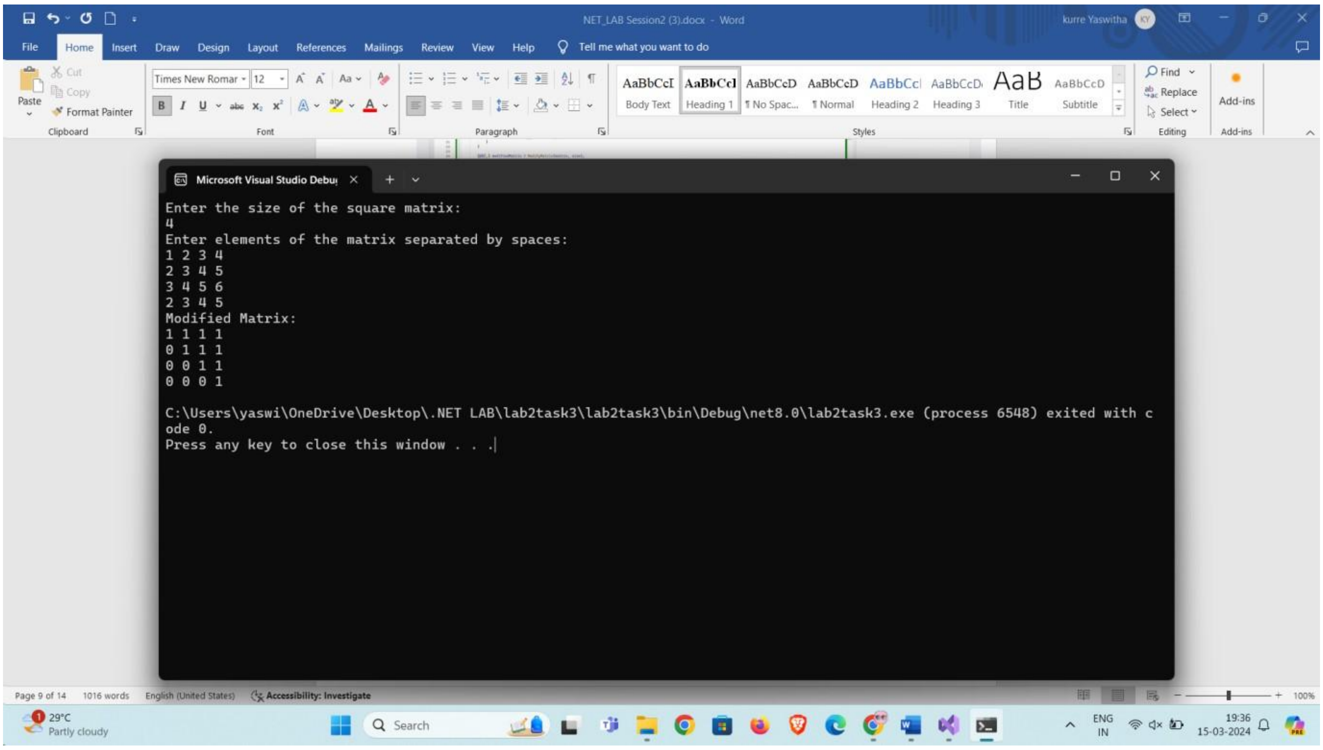
```
using System;

class Program
{
    static void Main()
    {
        Console.WriteLine("Enter the size of the square matrix:");
        int size = int.Parse(Console.ReadLine());

        int[,] matrix = new int[size, size];

        Console.WriteLine("Enter elements of the matrix separated by spaces:");

        for (int i = 0; i < size; i++)
        {
            string[] row = Console.ReadLine().Split(' ');

            for (int j = 0; j < size; j++)
            {
                matrix[i, j] = int.Parse(row[j]);
            }
        }

        int[,] modifiedMatrix = ModifyMatrix(matrix, size);

        Console.WriteLine("Modified Matrix:");
        PrintMatrix(modifiedMatrix, size);
    }

    static int[,] ModifyMatrix(int[,] matrix, int size)
    {
        int[,] modifiedMatrix = new int[size, size];

        for (int i = 0; i < size; i++)
        {
            for (int j = 0; j < size; j++)
            {
                if (j < i) // Elements to the left of the main diagonal
                {
                    modifiedMatrix[i, j] = 0;
                }
                else // Elements to the right of the main diagonal
                {
                    modifiedMatrix[i, j] = 1;
                }
            }
        }

        return modifiedMatrix;
```



```
    static int[,] ModifyMatrix(int[,] matrix, int size)
    {
        int[,] modifiedMatrix = new int[size, size];

        for (int i = 0; i < size; i++)
        {
            for (int j = 0; j < size; j++)
            {
                if (j < i) // Elements to the left of the main diagonal
                {
                    modifiedMatrix[i, j] = 0;
                }
                else // Elements to the right of the main diagonal
                {
                    modifiedMatrix[i, j] = 1;
                }
            }
        }

        return modifiedMatrix;
```

```
    }

    static void PrintMatrix(int[,] matrix, int size)
    {
        for (int i = 0; i < size; i++)
        {
            for (int j = 0; j < size; j++)
            {
                Console.Write(matrix[i, j] + " ");
            }
            Console.WriteLine();
        }
    }
}
```

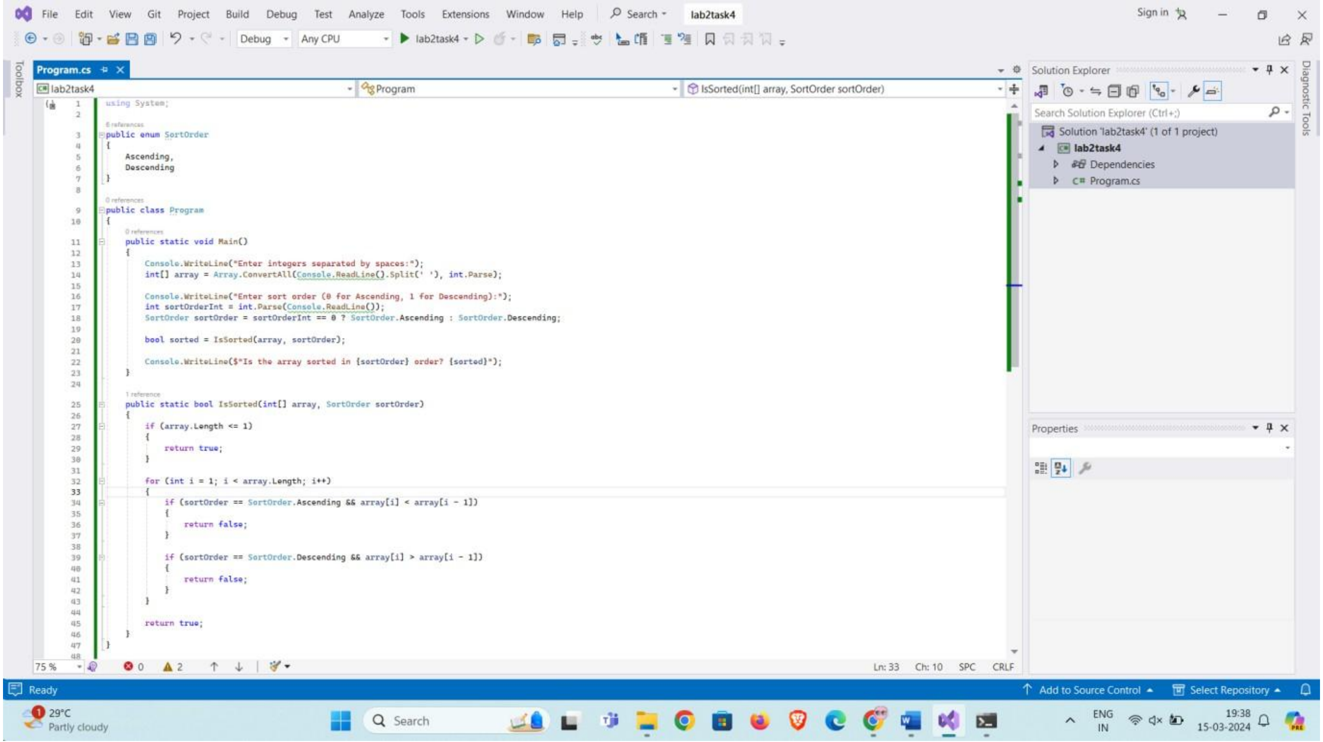
OUTPUT:



```
Enter the size of the square matrix:
4
Enter elements of the matrix separated by spaces:
1 2 3 4
2 3 4 5
3 4 5 6
2 3 4 5
Modified Matrix:
1 1 1 1
0 1 1 1
0 0 1 1
0 0 0 1

C:\Users\yaswi\OneDrive\Desktop\.NET LAB\lab2task3\lab2task3\bin\Debug\net8.0\lab2task3.exe (process 6548) exited with c
ode 0.
Press any key to close this window . . .
```

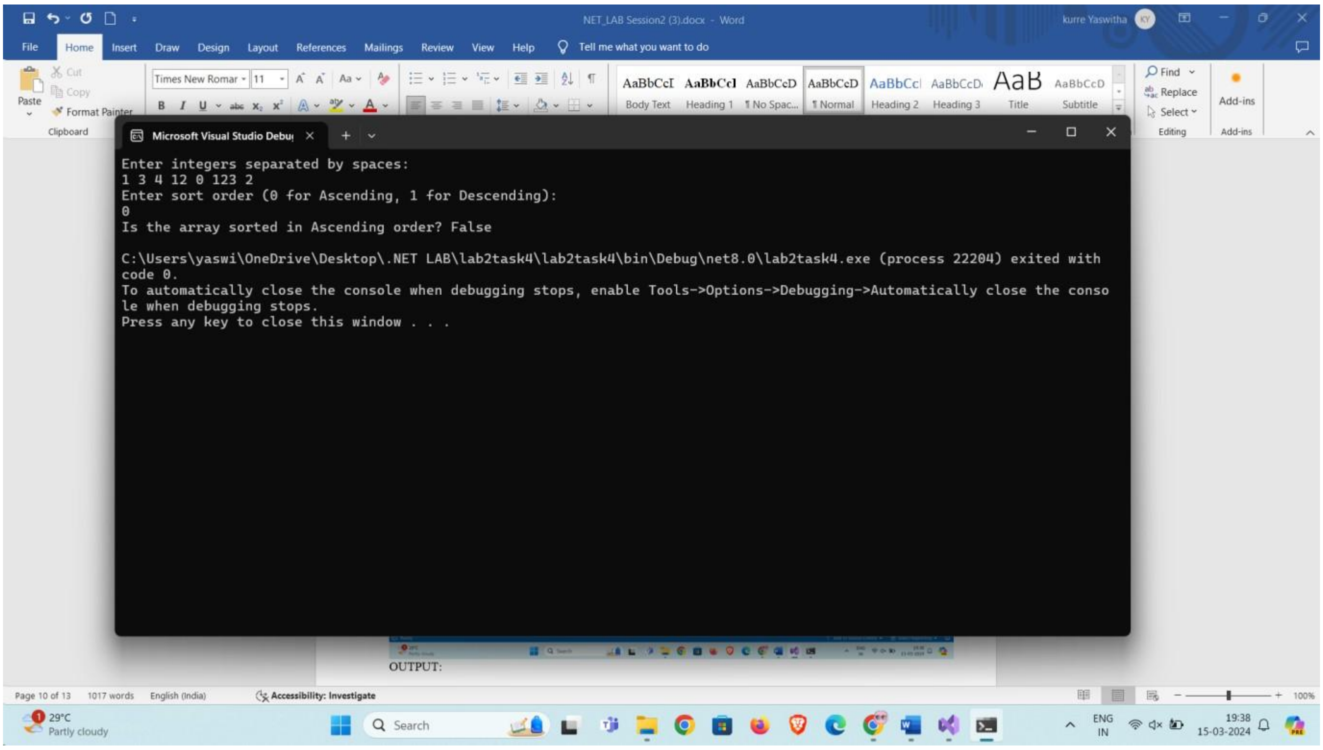
3. Write a C# code to implement the Tasks on Functions?

TASK 1: Create function *IsSorted*, determining whether a given *array* of integer values of arbitrary length is sorted in a given *order* (the order is set up by enum value *SortOrder*). Array and sort order are passed by parameters. Function does not change the array



```csharp
using System;

public enum SortOrder
{
    Ascending,
    Descending
}

public class Program
{
    public static void Main()
    {
        Console.WriteLine("Enter integers separated by spaces:");
        int[] array = Array.ConvertAll(Console.ReadLine().Split(' '), int.Parse);

        Console.WriteLine("Enter sort order (0 for Ascending, 1 for Descending):");
        int sortOrderInt = int.Parse(Console.ReadLine());
        SortOrder sortOrder = sortOrderInt == 0 ? SortOrder.Ascending : SortOrder.Descending;

        bool sorted = IsSorted(array, sortOrder);

        Console.WriteLine($"Is the array sorted in {sortOrder} order? {sorted}");
    }

    public static bool IsSorted(int[] array, SortOrder sortOrder)
    {
        if (array.Length <= 1)
        {
            return true;
        }

        for (int i = 1; i < array.Length; i++)
        {
            if (sortOrder == SortOrder.Ascending && array[i] < array[i - 1])
            {
                return false;
            }

            if (sortOrder == SortOrder.Descending && array[i] > array[i - 1])
            {
                return false;
            }
        }

        return true;
    }
}
```

OUTPUT:



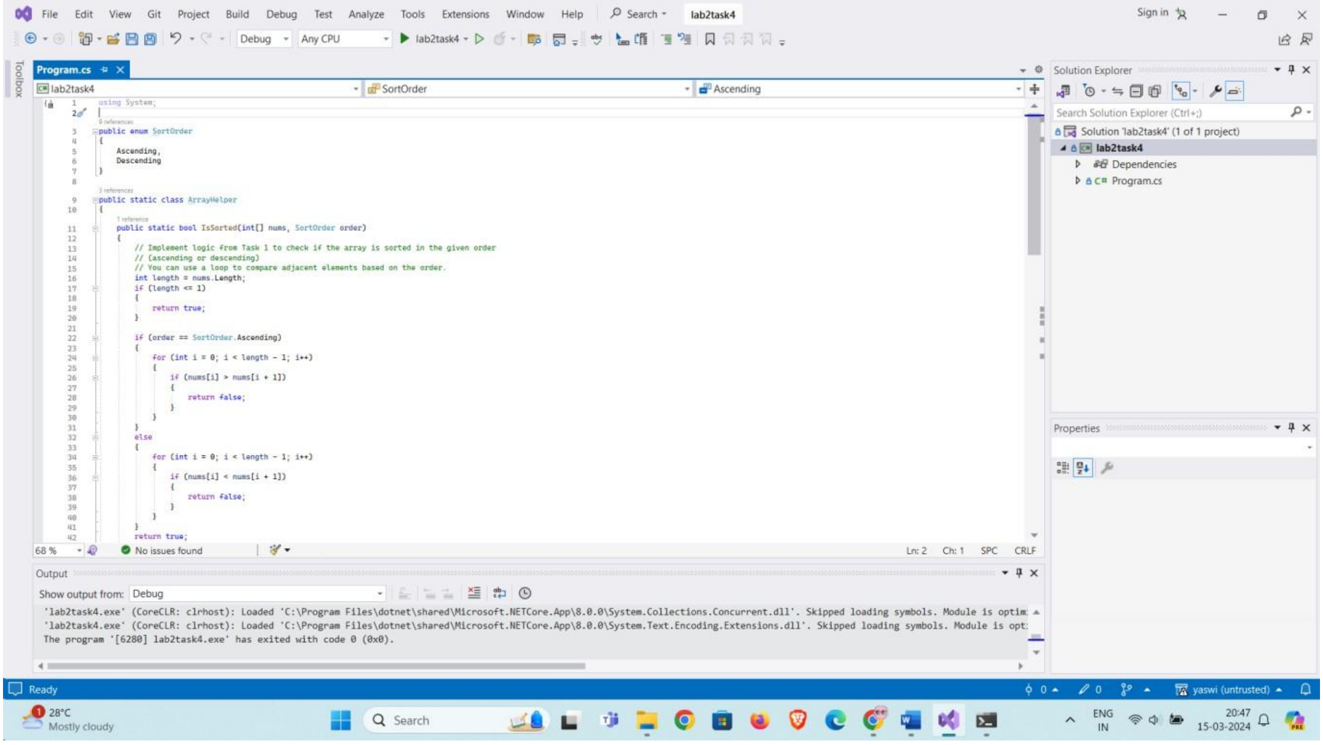
```
Enter integers separated by spaces:
1 3 4 12 0 123 2
Enter sort order (0 for Ascending, 1 for Descending):
0
Is the array sorted in Ascending order? False

C:\Users\yaswi\OneDrive\Desktop\.NET LAB\lab2task4\lab2task4\bin\Debug\net8.0\lab2task4.exe (process 22204) exited with
code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

OUTPUT:

TASK 2: Create function ***Transform***, replacing the value of each element of an integer ***array*** with the sum of this element value and its index, only if the given ***array*** is sorted in the given ***order*** (the order is set up by enum value ***SortOrder***). Array and sort order are passed by parameters. To check, if the array is sorted, the function ***IsSorted*** from the Task 1 is called.

Example

```
For {5, 17, 24, 88, 33, 2} and "ascending" sort order values in the array do not
change;
For {15, 10, 3} and "ascending" sort order values in the array do not change;
For {15, 10, 3} and "descending" sort order the values in the array change to {15,
11, 5}
```

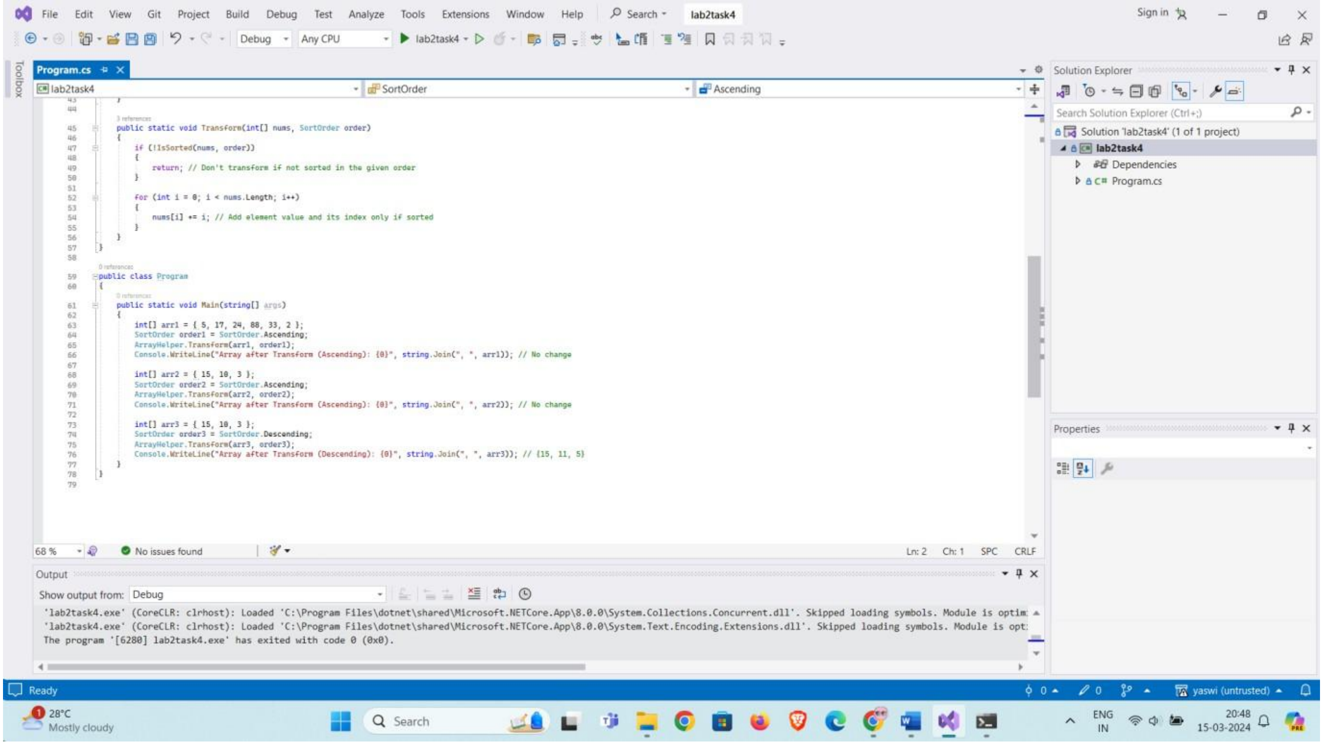


```
using System;

public enum SortOrder
{
    Ascending,
    Descending
}

public static class ArrayHelper
{
    public static bool IsSorted(int[] nums, SortOrder order)
    {
        // Implement logic from Task 1 to check if the array is sorted in the given order
        // (ascending or descending)
        // You can use a loop to compare adjacent elements based on the order.
        int length = nums.Length;
        if (length <= 1)
        {
            return true;
        }

        if (order == SortOrder.Ascending)
        {
            for (int i = 0; i < length - 1; i++)
            {
                if (nums[i] > nums[i + 1])
                {
                    return false;
                }
            }
        }
        else
        {
            for (int i = 0; i < length - 1; i++)
            {
                if (nums[i] < nums[i + 1])
                {
                    return false;
                }
            }
        }
        return true;
```

Output

Show output from: Debug

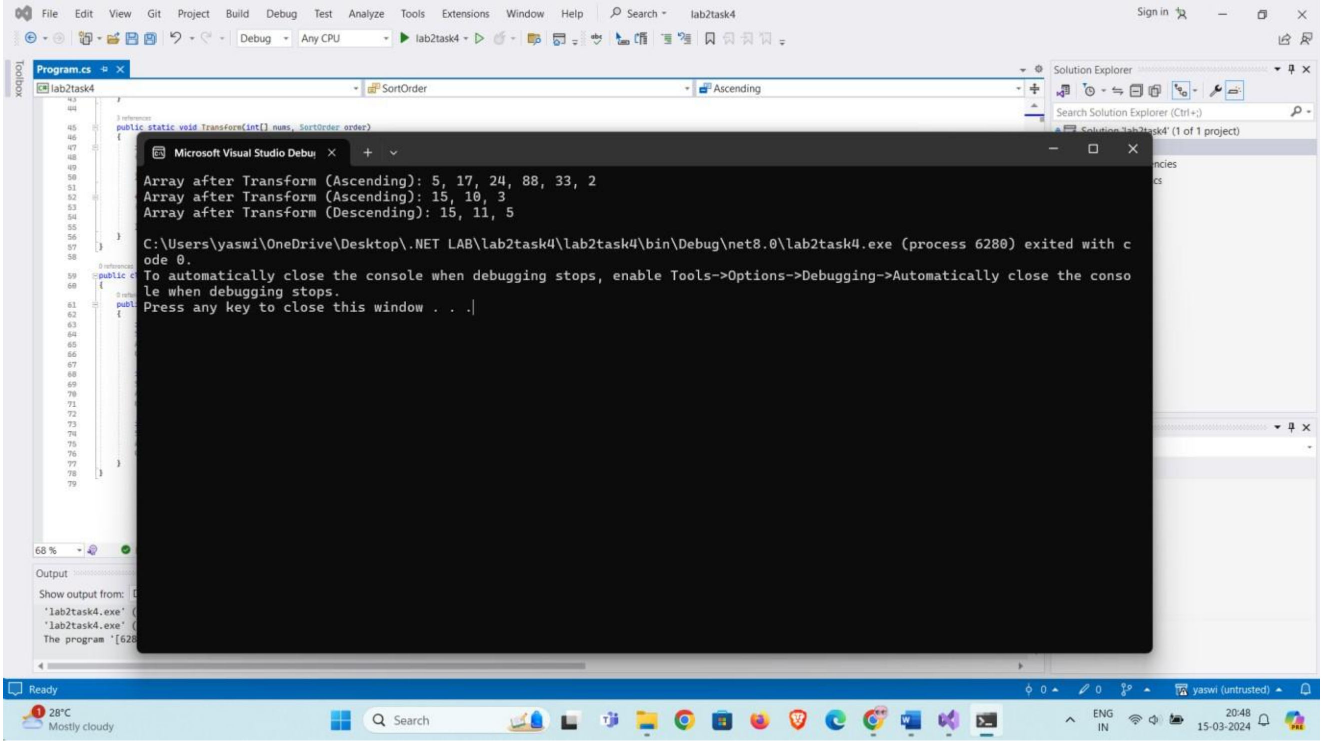
```
'lab2task4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Collections.Concurrent.dll'. Skipped loading symbols. Module is optim
'lab2task4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Text.Encoding.Extensions.dll'. Skipped loading symbols. Module is opt
The program '[6280] lab2task4.exe' has exited with code 0 (0x0).
```



```
    }

    public static void Transform(int[] nums, SortOrder order)
    {
        if (!IsSorted(nums, order))
        {
            return; // Don't transform if not sorted in the given order
        }

        for (int i = 0; i < nums.Length; i++)
        {
            nums[i] += i; // Add element value and its index only if sorted
        }
    }
}

public class Program
{
    public static void Main(string[] args)
    {
        int[] arr1 = { 5, 17, 24, 88, 33, 2 };
        SortOrder order1 = SortOrder.Ascending;
        ArrayHelper.Transform(arr1, order1);
        Console.WriteLine("Array after Transform (Ascending): {0}", string.Join(", ", arr1)); // No change

        int[] arr2 = { 15, 10, 3 };
        SortOrder order2 = SortOrder.Ascending;
        ArrayHelper.Transform(arr2, order2);
        Console.WriteLine("Array after Transform (Ascending): {0}", string.Join(", ", arr2)); // No change

        int[] arr3 = { 15, 10, 3 };
        SortOrder order3 = SortOrder.Descending;
        ArrayHelper.Transform(arr3, order3);
        Console.WriteLine("Array after Transform (Descending): {0}", string.Join(", ", arr3)); // {15, 11, 5}
    }
}
```

```
'lab2task4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Collections.Concurrent.dll'. Skipped loading symbols. Module is optim
'lab2task4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Text.Encoding.Extensions.dll'. Skipped loading symbols. Module is opt:
The program '[6280] lab2task4.exe' has exited with code 0 (0x0).
```

OUTPUT:

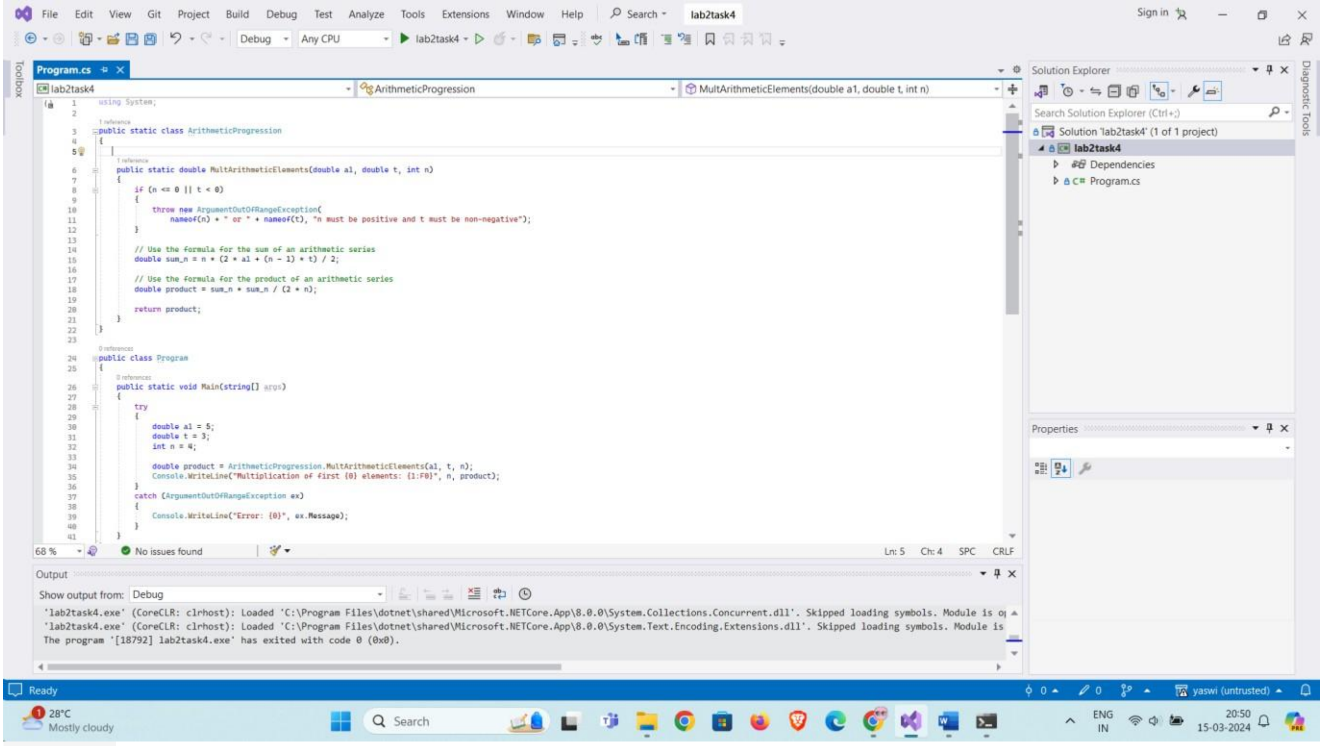


```
Array after Transform (Ascending): 5, 17, 24, 88, 33, 2
Array after Transform (Ascending): 15, 10, 3
Array after Transform (Descending): 15, 11, 5

C:\Users\yaswi\OneDrive\Desktop\.NET LAB\lab2task4\lab2task4\bin\Debug\net8.0\lab2task4.exe (process 6280) exited with c
ode 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

TASK 3: Create function ***MultArithmeticElements***, which determines the multiplication of a given number of first ***n*** elements of arithmetic progression of real numbers with a given initial element of progression ***a(1)*** and progression step ***t***. a(n) is calculated by the formula a(n+1) = a(n) + t.

Example

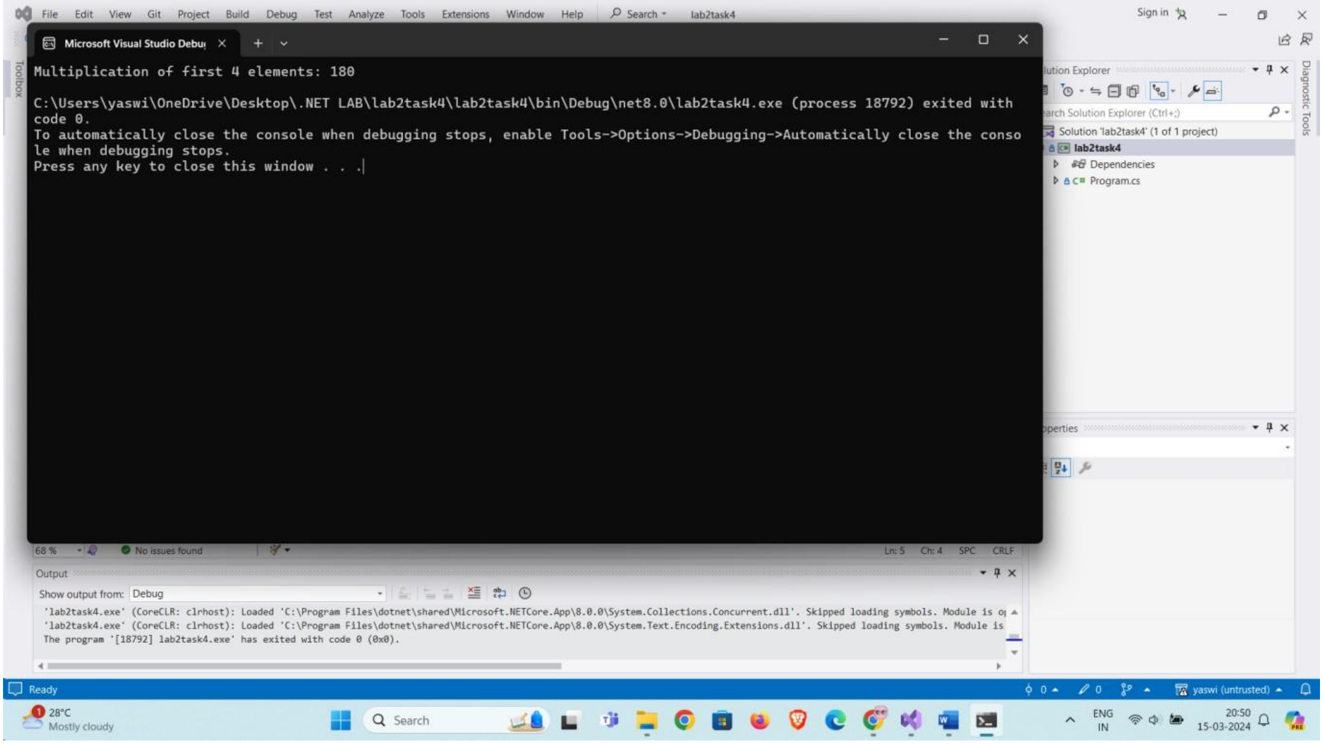
```
For a(1) = 5, t = 3, n = 4  multiplication equals to 5*8*11*14 = 6160
```



```
using System;

public static class ArithmeticProgression
{
    
    public static double MultArithmeticElements(double a1, double t, int n)
    {
        if (n <= 0 || t < 0)
        {
            throw new ArgumentOutOfRangeException(
                nameof(n) + " or " + nameof(t), "n must be positive and t must be non-negative");
        }

        // Use the formula for the sum of an arithmetic series
        double sum_n = n * (2 * a1 + (n - 1) * t) / 2;

        // Use the formula for the product of an arithmetic series
        double product = sum_n * sum_n / (2 * n);

        return product;
    }
}

public class Program
{
    public static void Main(string[] args)
    {
        try
        {
            double a1 = 5;
            double t = 3;
            int n = 4;

            double product = ArithmeticProgression.MultArithmeticElements(a1, t, n);
            Console.WriteLine("Multiplication of first {0} elements: {1:F0}", n, product);
        }
        catch (ArgumentOutOfRangeException ex)
        {
            Console.WriteLine("Error: {0}", ex.Message);
        }
    }
```

```
'lab2task4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Collections.Concurrent.dll'. Skipped loading symbols. Module is o
'lab2task4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Text.Encoding.Extensions.dll'. Skipped loading symbols. Module is
The program '[18792] lab2task4.exe' has exited with code 0 (0x0).
```

OUTPUT:

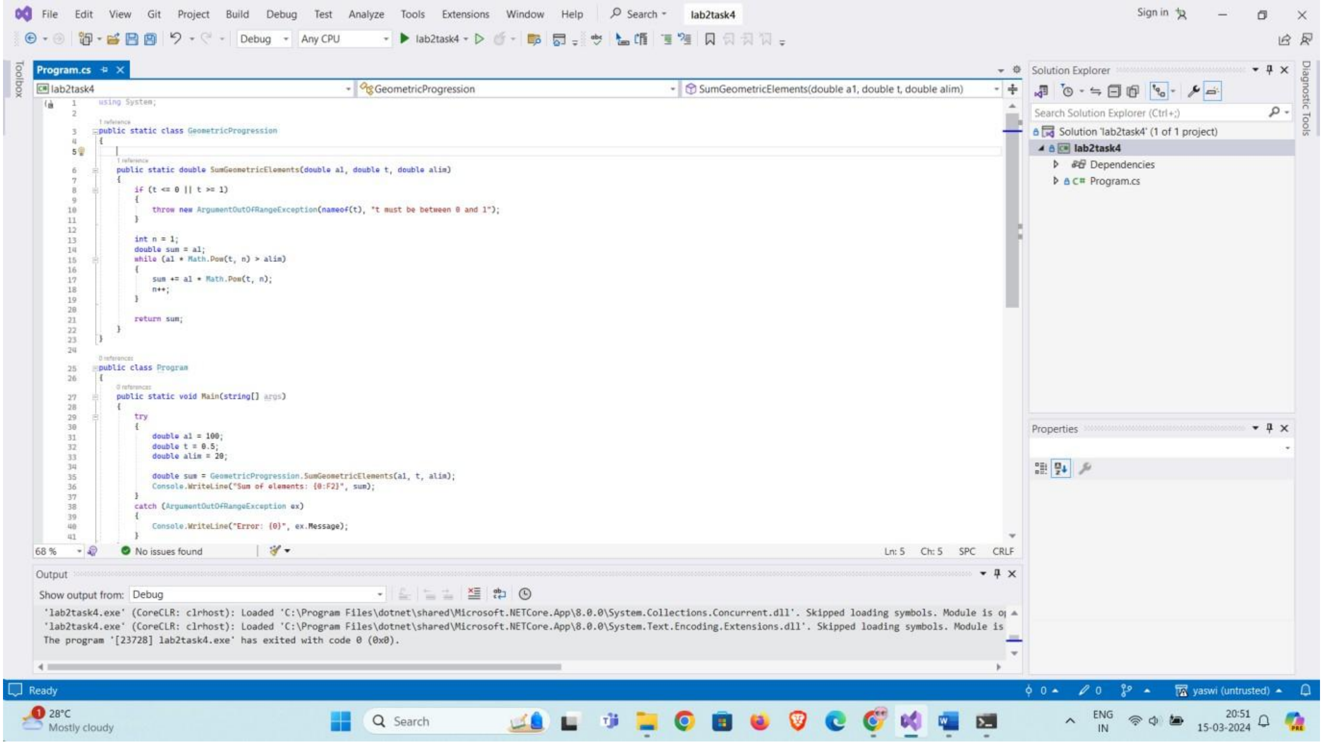


```
Multiplication of first 4 elements: 180

C:\Users\yaswi\OneDrive\Desktop\.NET LAB\lab2task4\lab2task4\bin\Debug\net8.0\lab2task4.exe (process 18792) exited with
code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

TASK 4: Create function ***SumGeometricElements***, determining the sum of the first elements of a decreasing geometric progression of real numbers with a given initial element of a progression ***a(1)***) and a given progression step ***t***, while the last element must be greater than a given ***alim***. an is calculated by the formula a(n+1) = a(n) * t, 0<t<1.

Example

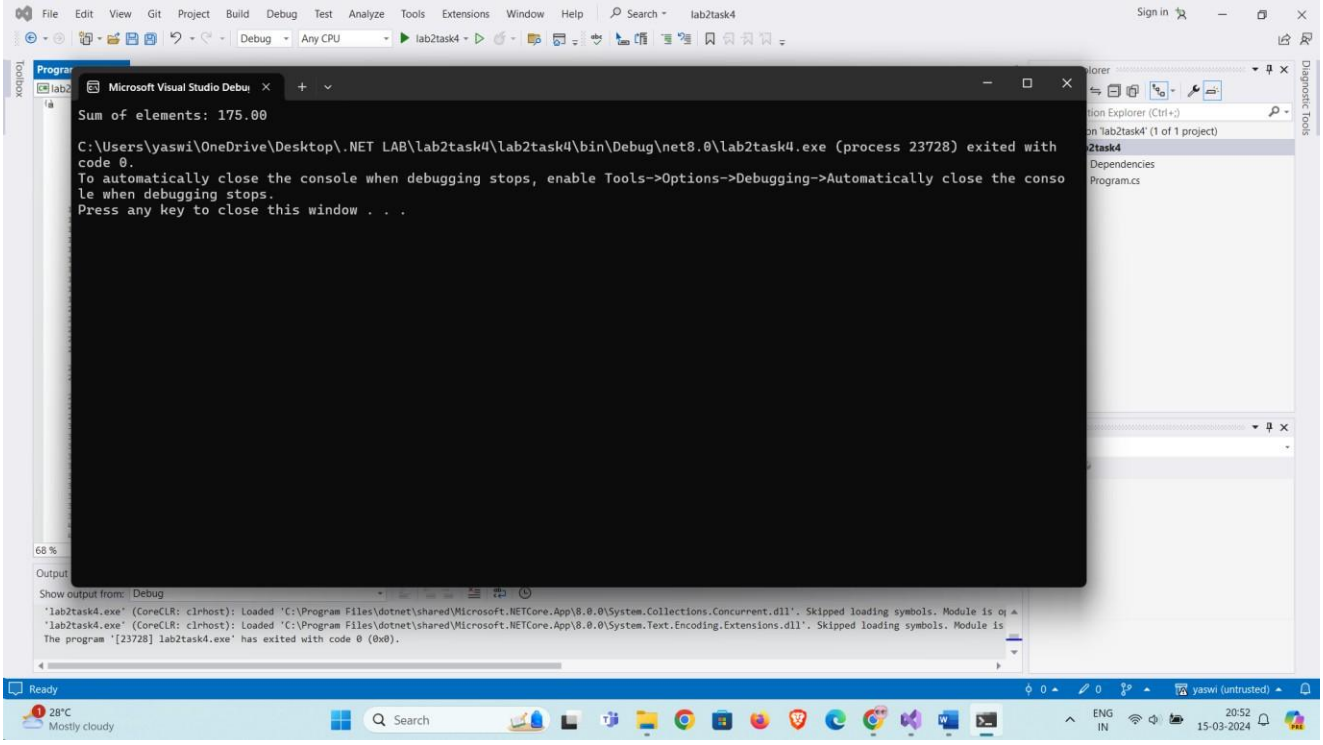
```
For a progression, where a(1) = 100, and t = 0.5, the sum of the
first elements, grater than alim = 20, equals to 100+50+25 = 175
```



```
using System;

public static class GeometricProgression
{

    public static double SumGeometricElements(double a1, double t, double alim)
    {
        if (t <= 0 || t >= 1)
        {
            throw new ArgumentOutOfRangeException(nameof(t), "t must be between 0 and 1");
        }

        int n = 1;
        double sum = a1;
        while (a1 * Math.Pow(t, n) > alim)
        {
            sum += a1 * Math.Pow(t, n);
            n++;
        }

        return sum;
    }
}

public class Program
{
    public static void Main(string[] args)
    {
        try
        {
            double a1 = 100;
            double t = 0.5;
            double alim = 20;

            double sum = GeometricProgression.SumGeometricElements(a1, t, alim);
            Console.WriteLine("Sum of elements: {0:F2}", sum);
        }
        catch (ArgumentOutOfRangeException ex)
        {
            Console.WriteLine("Error: {0}", ex.Message);
        }
```

```
'lab2task4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Collections.Concurrent.dll'. Skipped loading symbols. Module is o
'lab2task4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Text.Encoding.Extensions.dll'. Skipped loading symbols. Module is
The program '[23728] lab2task4.exe' has exited with code 0 (0x0).
```

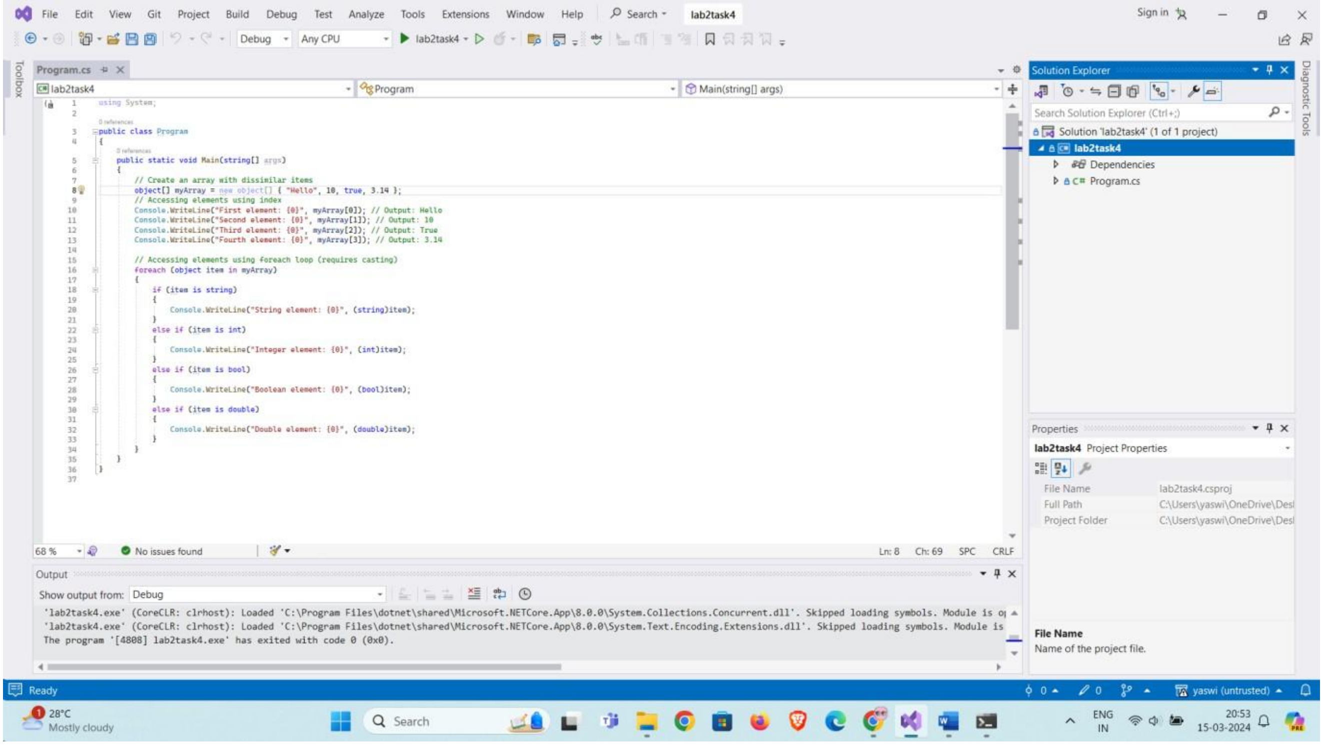
OUTPUT:



```
Sum of elements: 175.00

C:\Users\yaswi\OneDrive\Desktop\.NET LAB\lab2task4\lab2task4\bin\Debug\net8.0\lab2task4.exe (process 23728) exited with
code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

POST-LAB

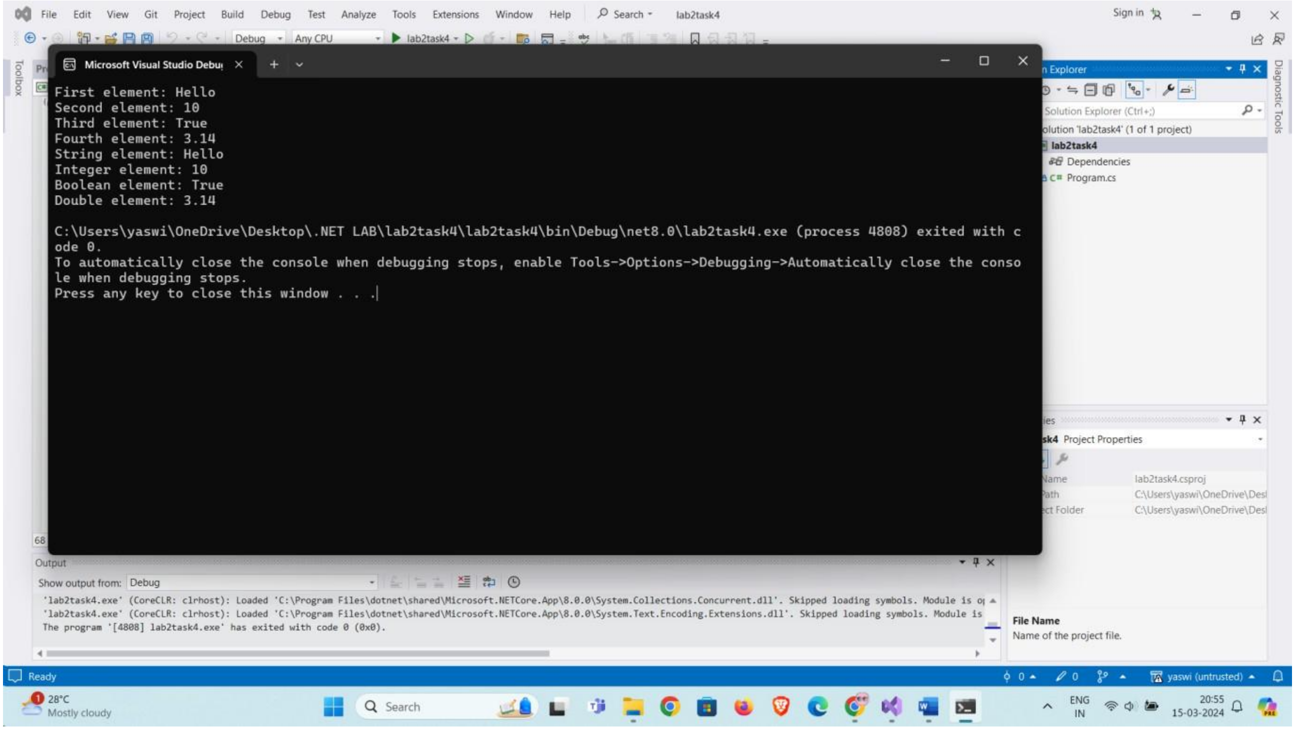
1. Construct an Array with dissimilar items and access it?



```
using System;

public class Program
{
    public static void Main(string[] args)
    {
        // Create an array with dissimilar items
        object[] myArray = new object[] { "Hello", 10, true, 3.14 };
        // Accessing elements using index
        Console.WriteLine("First element: {0}", myArray[0]); // Output: Hello
        Console.WriteLine("Second element: {0}", myArray[1]); // Output: 10
        Console.WriteLine("Third element: {0}", myArray[2]); // Output: True
        Console.WriteLine("Fourth element: {0}", myArray[3]); // Output: 3.14

        // Accessing elements using foreach loop (requires casting)
        foreach (object item in myArray)
        {
            if (item is string)
            {
                Console.WriteLine("String element: {0}", (string)item);
            }
            else if (item is int)
            {
                Console.WriteLine("Integer element: {0}", (int)item);
            }
            else if (item is bool)
            {
                Console.WriteLine("Boolean element: {0}", (bool)item);
            }
            else if (item is double)
            {
                Console.WriteLine("Double element: {0}", (double)item);
            }
        }
    }
}
```

```
'lab2task4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Collections.Concurrent.dll'. Skipped loading symbols. Module is o
'lab2task4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Text.Encoding.Extensions.dll'. Skipped loading symbols. Module is
The program '[4808] lab2task4.exe' has exited with code 0 (0x0).
```

OUTPUT:



```
First element: Hello
Second element: 10
Third element: True
Fourth element: 3.14
String element: Hello
Integer element: 10
Boolean element: True
Double element: 3.14

C:\Users\yaswi\OneDrive\Desktop\.NET LAB\lab2task4\lab2task4\bin\Debug\net8.0\lab2task4.exe (process 4808) exited with c
ode 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

```
'lab2task4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Collections.Concurrent.dll'. Skipped loading symbols. Module is o
'lab2task4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Text.Encoding.Extensions.dll'. Skipped loading symbols. Module is
The program '[4808] lab2task4.exe' has exited with code 0 (0x0).
```