

Lab 2: Tasks on Looping, Arrays and Functions

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)

$n = 246$ $result = 0$

answer1:

The screenshot shows the Microsoft Visual Studio IDE. On the left, the 'Lab2task1.cs' file is open, displaying the following C# code:

```

6  public static void Main(String[] args)
7  {
8      int n1 = 1234;
9      int result1 = CalculateSumOfOddDigits(n1);
10
11     int n2 = 246;
12     int result2 = CalculateSumOfOddDigits(n2);
13
14     Console.WriteLine("For n = 1234, the result is: " + result1);
15     Console.WriteLine("For n = 246, the result is: " + result2);
16 }
17
18 static int CalculateSumOfOddDigits(int n)
19 {
20     int sum = 0;
21     string str = n.ToString();
22     foreach (char digit in str)
23     {
24         int digitInt = int.Parse(digit.ToString());
25         if (digitInt % 2 != 0)
26         {
27             sum += digitInt;
28         }
29     }
30     return sum;
31 }
32
33
34

```

On the right, the 'Console' window shows the output of the program:

```

For n = 1234, the result is: 4
For n = 246, the result is: 0
C:\Users\Sambu\source\repos\lab2\Lab2task1\Lab2task1\bin\Debug\net8.0\Lab2task1.exe (process 12160) exited with code 0.
Press any key to close this window . . .

```

The bottom status bar indicates 'Ln: 28 Ch: 30'.

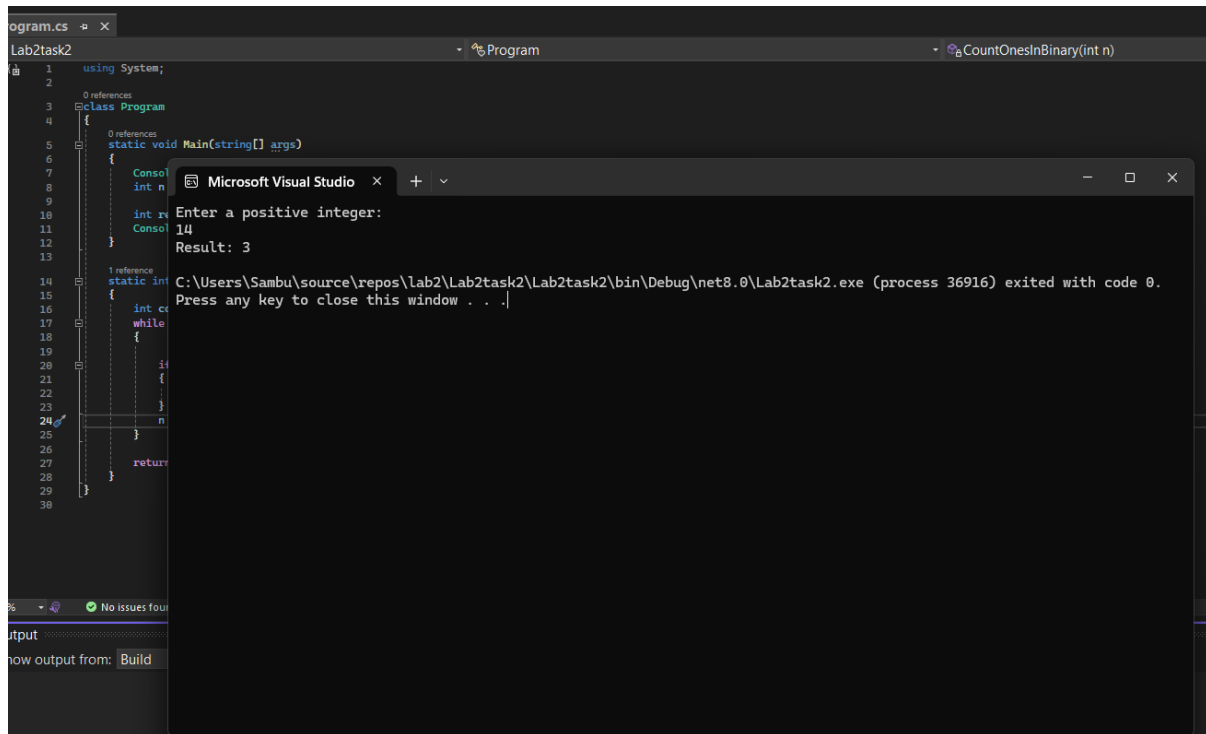
TASK2: For a positive integer n calculate the result value, which is equal to the sum of the “1” in the binary representation of n .

Example

$n = 14$ (decimal) = 1110 (binary) $result = 3$

$n = 128$ (decimal) = 1000 0000 (binary) $result = 1$

answer2:



The screenshot shows the Microsoft Visual Studio IDE. On the left, the 'Lab2task2.cs' file is open, displaying a C# program. The code includes a `using System;` statement, a `Program` class, and a `Main` method that takes a string array `args`. Inside the `Main` method, there is a `Console` class with a `WriteLine` method that prints the result of a calculation. The code also includes a `while` loop and a `return` statement. On the right, the 'Console' window is open, showing the output of the program. The output indicates that the program entered a positive integer of 14 and calculated a result of 3. The console window also shows the path to the executable file and the exit code.

```
1 using System;
2
3 0 references
4 class Program
5 {
6     0 references
7     static void Main(string[] args)
8     {
9         Console
10        int n
11        Enter a positive integer:
12        Console
13        14
14        Result: 3
15
16        1 reference
17        static int
18        {
19            int r
20            while
21            {
22                if
23                {
24                    n
25                }
26            }
27            return
28        }
29    }
30 }
```

C:\Users\Sambu\source\repos\lab2\Lab2task2\Lab2task2\bin\Debug\net8.0\Lab2task2.exe (process 36916) exited with code 0.
Press any key to close this window . . .

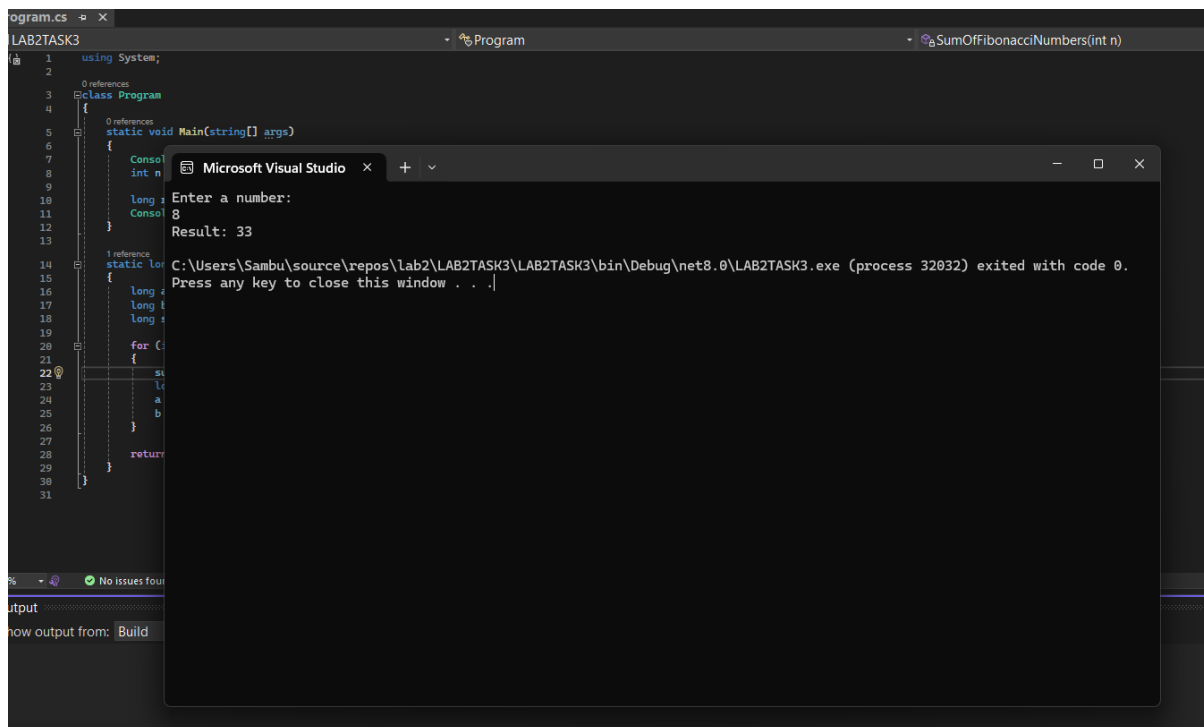
TASK3: For a positive integer n , calculate the result value equal to the sum of the first n Fibonacci numbers. Note: Fibonacci numbers are a series of numbers in which each next number is equal to the sum of the two preceding ones: 0, 1, 1, 2, 3, 5, 8, 13... ($F_0=0$, $F_1=F_2=1$, then $F(n)=F(n-1)+F(n-2)$ for $n>2$)

Example

`n = 8 result = 33`

`n = 11 result = 143`

answer3:



1. 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

$$\{10, 5, 3, 4\} \Rightarrow \{4, 5, 3, 10\}$$
$$\{100, 2, 3, 4, 5\} \Rightarrow \{100, 4, 3, 2, 5\}$$

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

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
```

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}}
```

Answer1:

```
ask1.1
{
    4 {
    5     0 references
    6     static void Main(string[] args)
    7     {
    8         Console.WriteLine("Enter :");
    9         string input = Console.ReadLine();
   10         string[] numbers = input.Split(' ');
   11         int[] nums = new int[numbers.Length];
   12         for (int i = 0; i < numbers.Length; i++)
   13         {
   14             nums[i] = int.Parse(numbers[i]);
   15         }
   16         SwapEvenElements(nums);
   17         Console.WriteLine("Result:");
   18         foreach (int num in nums)
   19         {
   20             Console.Write(num + " ");
   21         }
   22     }
   23 }
   24
   25 1 reference
   26 static void SwapEvenElements(int[] nums)
   27 {
   28     int left = 0;
   29     int right = nums.Length - 1;
   30     while (left < right)
   31     {
   32         if (nums[left] % 2 == 0 && nums[right] % 2 == 0)
   33         {
   34             int temp = nums[left];
   35             nums[left] = nums[right];
   36             nums[right] = temp;
   37         }
   38         left++;
   39         right--;
   40     }
   41 }
}

Microsoft Visual Studio x + v
Enter :
10 5 3 4
Result:
4 5 3 10
C:\Users\Sambu\source\repos\lab2\lab2task1.1\bin\Debug\net8.0\lab2task1.1.exe (pro
de 0.
Press any key to close this window . . |

Output from: Build
```

Answer2:

The screenshot shows a Visual Studio IDE with a C# file named `Program.cs` and a console window. The code defines a `Program` class with a `Main` method that reads an input string, splits it into an array of integers, and calls a static method `DistanceBetweenFirstAndLastMax`. This method finds the first and last indices of the maximum value in the array and returns the absolute difference between them. The console output shows the input `4 100 3 4` and the result `0`.

```
1 using System;
2 using System.Linq;
3
4 class Program
5 {
6     static void Main(string[] args)
7     {
8         Console.WriteLine("Enter :");
9         string input = Console.ReadLine();
10        string[] numbers = input.Split(' ');
11
12        int[] nums = new int[numbers.Length];
13        for (int i = 0; i < numbers.Length; i++)
14        {
15            nums[i] = int.Parse(numbers[i]);
16        }
17
18        int result = DistanceBetweenFirstAndLastMax(nums);
19        Console.WriteLine("Result: " + result);
20    }
21
22    static int DistanceBetweenFirstAndLastMax(int[] nums)
23    {
24        int max = nums.Max();
25        int firstMaxIndex = Array.IndexOf(nums, max);
26        int lastMaxIndex = Array.LastIndexOf(nums, max);
27
28        return Math.Abs(lastMaxIndex - firstMaxIndex);
29    }
30 }
31
```

Microsoft Visual Studio

Enter :
4 100 3 4
Result: 0

C:\Users\Sambu\source\repos\lab2\lab2task2.1\lab2task2.1\bin\Debug\net8.0\lab2task2.1.exe (process 3...)
Press any key to close this window . . .

Answer3:

The screenshot shows a Visual Studio IDE with a C# file named `Program.cs` and a console window. The code defines a `Program` class with a `Main` method that initializes a 2D array `matrix` and calls a static method `TransformMatrix`. This method iterates through the matrix and applies a transformation based on the value of `j` relative to `i`. The console output shows the transformed matrix.

```
1 using System;
2
3 class Program
4 {
5     static void Main(string[] args)
6     {
7         int[,] matrix = {
8             {2, 4, 3, 3},
9             {5, 7, 8, 5},
10            {2, 4, 3, 3},
11            {5, 7, 8, 5}
12        };
13
14        TransformMatrix(matrix);
15
16        for (int i = 0; i < matrix.GetLength(0); i++)
17        {
18            for (int j = 0; j < matrix.GetLength(1); j++)
19            {
20                Console.Write(matrix[i, j] + " ");
21            }
22            Console.WriteLine();
23        }
24    }
25
26    static void TransformMatrix(int[,] matrix)
27    {
28        int n = matrix.GetLength(0);
29
30        for (int i = 0; i < n; i++)
31        {
32            for (int j = 0; j < n; j++)
33            {
34                if (j < i)
35                    matrix[i, j] = 0;
36                else if (j > i)
37                    matrix[i, j] = 0;
38            }
39        }
40    }
41 }
42
```

Microsoft Visual Studio

2 1 1 1
0 7 1 1
0 0 3 1
0 0 0 5

C:\Users\Sambu\source\repos\lab2\lab2task3.1\lab2task3.1\bin\Debug\net8.0\lab2task3.1.exe (process 3...)
Press any key to close this window . . .

1. 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

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}

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

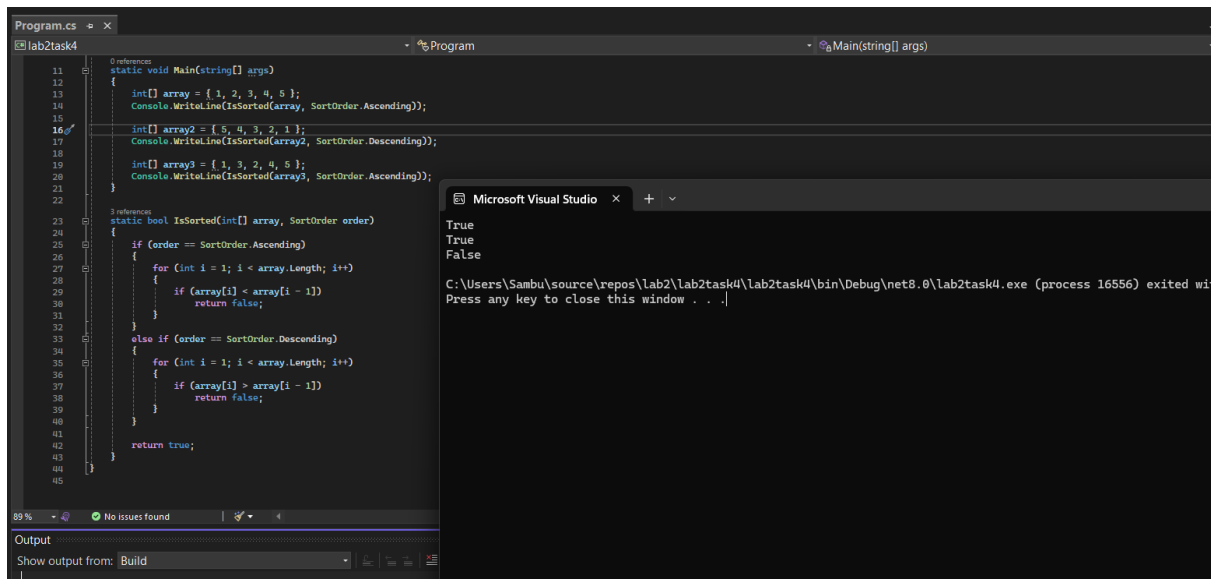
For $a(1) = 5$, $t = 3$, $n = 4$ multiplication equals to $5*8*11*14 = 6160$

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 $100+50+25 = 175$

Answer1:



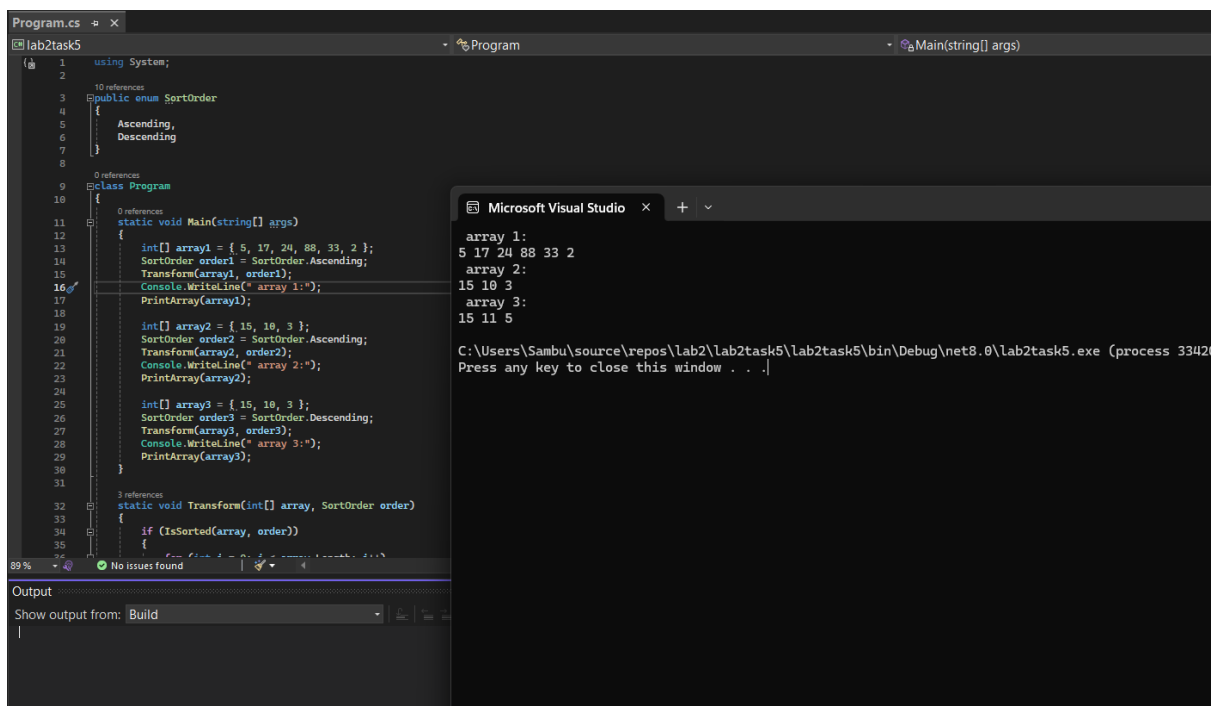
```
Program.cs
lab2task4

11 0 references
12 static void Main(string[] args)
13 {
14     int[] array = { 1, 2, 3, 4, 5 };
15     Console.WriteLine(IsSorted(array, SortOrder.Ascending));
16
17     int[] array2 = { 5, 4, 3, 2, 1 };
18     Console.WriteLine(IsSorted(array2, SortOrder.Descending));
19
20     int[] array3 = { 1, 3, 2, 4, 5 };
21     Console.WriteLine(IsSorted(array3, SortOrder.Ascending));
22 }
23
24 3 references
25 static bool IsSorted(int[] array, SortOrder order)
26 {
27     if (order == SortOrder.Ascending)
28     {
29         for (int i = 1; i < array.Length; i++)
30         {
31             if (array[i] < array[i - 1])
32                 return false;
33         }
34     }
35     else if (order == SortOrder.Descending)
36     {
37         for (int i = 1; i < array.Length; i++)
38         {
39             if (array[i] > array[i - 1])
40                 return false;
41         }
42     }
43     return true;
44 }
45

Microsoft Visual Studio
True
True
False

C:\Users\Sambu\source\repos\lab2\lab2task4\lab2task4\bin\Debug\net8.0\lab2task4.exe (process 16556) exited wi
Press any key to close this window . . .
```

Answer2:



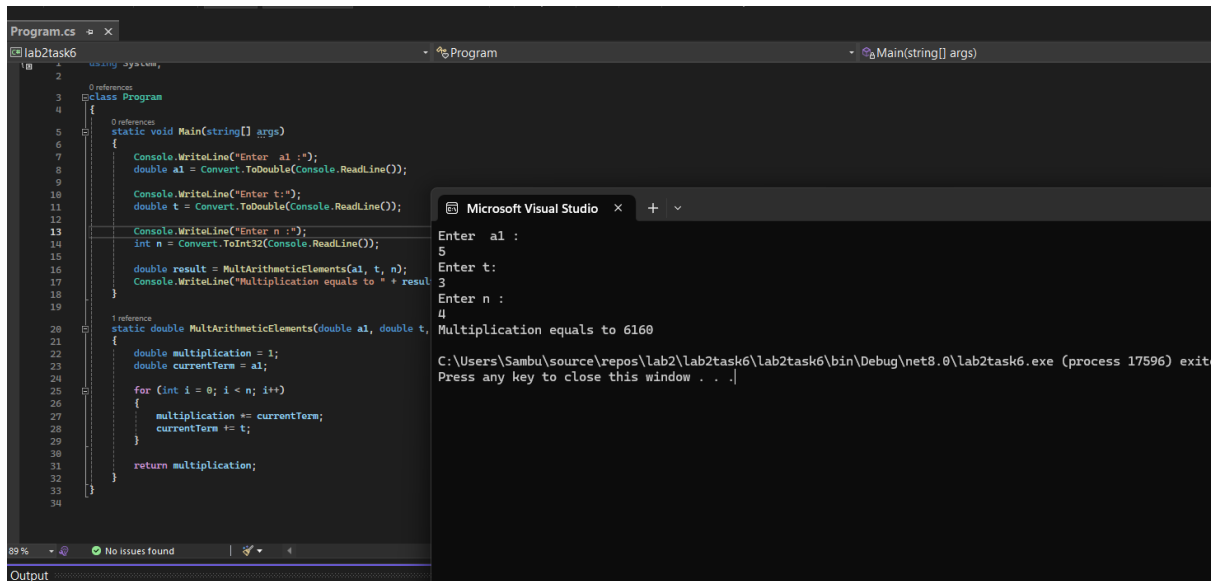
```
Program.cs
lab2task5

1  using System;
2
3 10 references
4  public enum SortOrder
5  {
6      Ascending,
7      Descending
8  }
9
10 0 references
11 class Program
12 {
13     0 references
14     static void Main(string[] args)
15     {
16         int[] array1 = { 5, 17, 28, 88, 33, 2 };
17         SortOrder order1 = SortOrder.Ascending;
18         Transform(array1, order1);
19         Console.WriteLine(" array 1:");
20         PrintArray(array1);
21
22         int[] array2 = { 15, 18, 3 };
23         SortOrder order2 = SortOrder.Ascending;
24         Transform(array2, order2);
25         Console.WriteLine(" array 2:");
26         PrintArray(array2);
27
28         int[] array3 = { 15, 18, 3 };
29         SortOrder order3 = SortOrder.Descending;
30         Transform(array3, order3);
31         Console.WriteLine(" array 3:");
32         PrintArray(array3);
33     }
34
35 3 references
36 static void Transform(int[] array, SortOrder order)
37 {
38     if (IsSorted(array, order))
39     {
40     }
41 }
42
43 2.6
44
45 89% No issues found

Microsoft Visual Studio
array 1:
5 17 24 88 33 2
array 2:
15 18 3
array 3:
15 11 5

C:\Users\Sambu\source\repos\lab2\lab2task5\lab2task5\bin\Debug\net8.0\lab2task5.exe (process 3342)
Press any key to close this window . . .
```

Answer3:



The screenshot shows the Visual Studio IDE with a C# program named `lab2task6`. The code defines a `Program` class with a `Main` method that prompts the user for three inputs: `a1`, `t`, and `n`. It then calculates the product of `a1` and `t` raised to the power of `n` using the `MultArithmeticElements` method. The output window shows the user entering `5` for `a1`, `3` for `t`, and `4` for `n`, resulting in the output `Multiplication equals to 6160`.

```
Program.cs
lab2task6
using System;

class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Enter a1 :");
        double a1 = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("Enter t:");
        double t = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("Enter n :");
        int n = Convert.ToInt32(Console.ReadLine());

        double result = MultArithmeticElements(a1, t, n);
        Console.WriteLine("Multiplication equals to " + result);
    }

    static double MultArithmeticElements(double a1, double t,
    {
        double multiplication = 1;
        double currentTerm = a1;

        for (int i = 0; i < n; i++)
        {
            multiplication *= currentTerm;
            currentTerm *= t;
        }

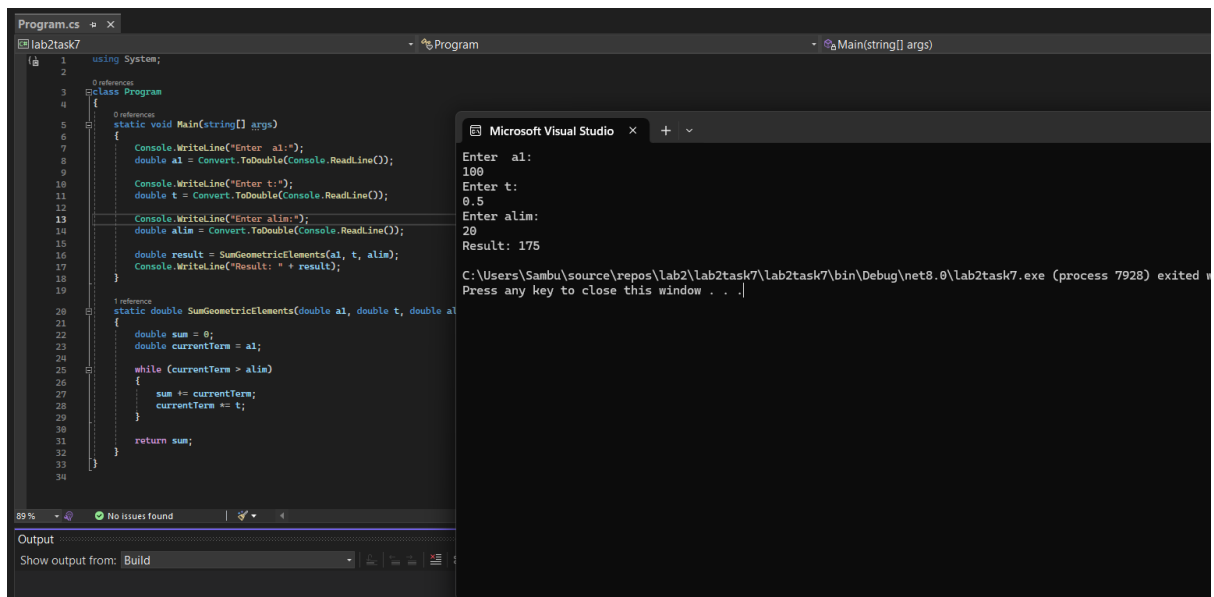
        return multiplication;
    }
}
```

Microsoft Visual Studio

Enter a1 :
5
Enter t:
3
Enter n :
4
Multiplication equals to 6160

C:\Users\Sambu\source\repos\lab2\lab2task6\lab2task6\bin\Debug\net8.0\lab2task6.exe (process 17596) exited with code 0.
Press any key to close this window . . .

Answer4:



The screenshot shows the Visual Studio IDE with a C# program named `lab2task7`. The code defines a `Program` class with a `Main` method that prompts the user for three inputs: `a1`, `t`, and `alim`. It then calculates the sum of the geometric series $a1 + a1 \cdot t + a1 \cdot t^2 + \dots + a1 \cdot t^{alim-1}$ using the `SumGeometricElements` method. The output window shows the user entering `100` for `a1`, `0.5` for `t`, and `20` for `alim`, resulting in the output `Result: 175`.

```
Program.cs
lab2task7
using System;

class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Enter a1:");
        double a1 = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("Enter t:");
        double t = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("Enter alim:");
        double alim = Convert.ToDouble(Console.ReadLine());

        double result = SumGeometricElements(a1, t, alim);
        Console.WriteLine("Result: " + result);
    }

    static double SumGeometricElements(double a1, double t, double alim)
    {
        double sum = 0;
        double currentTerm = a1;

        while (currentTerm > 0)
        {
            sum += currentTerm;
            currentTerm *= t;
        }

        return sum;
    }
}
```

Microsoft Visual Studio

Enter a1:
100
Enter t:
0.5
Enter alim:
20
Result: 175

C:\Users\Sambu\source\repos\lab2\lab2task7\lab2task7\bin\Debug\net8.0\lab2task7.exe (process 7928) exited with code 0.
Press any key to close this window . . .