**COMSATS UNIVERSITY ISLAMBAD ,ATTOCK CAMPUS**

**LAB ASSIGNMENT NO 1**

****

**SUBMITTED BY : Nida Eman**

**REGSTRATION NO : FA21-BCS-002**

**SUBMITTED TO : SIR Syed Bilal Haider**

**Subject : compiler construction (Lab)**

**DATE : 13/sep/2024**

**Topic : C# Regex**

**Defination :**

Regular Expression (Regex) is a pattern that will help in checking if a certain string fits the said type of pattern. Some of the functions can include searching for text, extracting and replacing text within the document or any other task that you may require of a computer.

e. g : The pattern which we have set for the Regex is “ ^m. t$”.

* ^ symbol points to the fact that string can begin only with the character m.
* . refers to any one letter or character consonant or vowel.
* In the symbol >, $ stand for string ends with**.**

**Syntax :**

Here are some basic regex elements:Here are some basic regex elements:

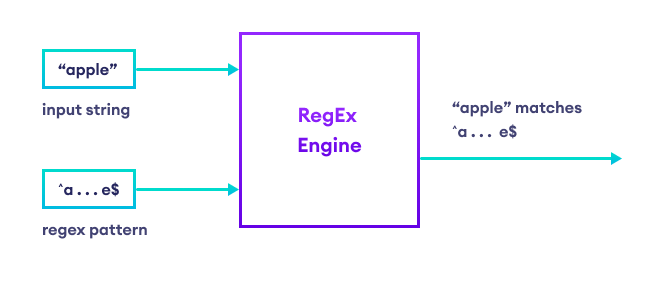
* \*: Means the same as 0 and represents the number of times the preceding element must occur.
* + : Period One or more repetitions of the preceding element.
* ? : Matches zero or one of the preceding element.
* {n} : Assembles n repetitions that matches exactly the preceding configuration.
* {n,} : Matches n or more repetitions of the last used element.
* {n,m} : Same elements paired up between n – 1 and m – 1 repetitions of the preceding element.

**Special Chracters in Regex:**

* \d : Matches any digit (equivalent to [0-9]).
* \D : Matches any non-digit.
* \w : Matches any word character (alphanumeric + underscore).
* \W : Matches any non-word character.
* \s : Matches any whitespace character (space, tab, newline, etc.).
* \S : Matches any non-whitespace character.

**How Regex works in C#?**

In C#, there is an engine called regex engine which internally checks the regex pattern in the given string.



**Regex Methods :**

Here are the different methods used in regex:

* Regex. IsMatch(string input, string pattern): It will return true if it finds that the string is a match for the pattern we have specified.
* Regex. Match(string input, string pattern): It does not look for all of the parameter from the Table but rather only seeks to return the one value that matches the parameter from the Table.
* Regex. Matches(string input, string pattern): They all return all of the matches of the search.
* Regex. Replace(string input, string pattern, string replacement): Uses a replacement string and replaces all the matches found in a specified string.
* Regex. Split(string input, string pattern): Divides the input data in accordance with the indicated pattern.

**Examples :**

For verifying the email will use the following method :

string pattern = @"^[\w\.-]+@[\w\.-]+\.\w{2,}$";

bool isValid = Regex.IsMatch("nidaeman@gmail.com", pattern);

**Performance Consideration:**

* And, off course, regex is powerful, however complex pattern can be very slow on large input strings.
* We should not use the patterns containing much backtracking since it will have severe performance problem.
* Compile regex should be used if performance is critical in your case you should specify RegexOptions. Compiled.

**Code Example:**

using System;

using System.Text.RegularExpressions;

class Program

{

static void Main()

{

string email = "test@example.com";

string pattern = @"^[\w\.-][+@[\w\.-]+\.\w{2,}$](mailto:+@[\w\.-%5d+\.\w%7b2,%7d$)"; // Regex pattern for validating email

bool isValid = Regex.IsMatch(email, pattern);

if (isValid) //ouput validation

{

Console.WriteLine($"{email} valid email address");

}

else

{

Console.WriteLine($"{email} is NOT a valid email address.");

}

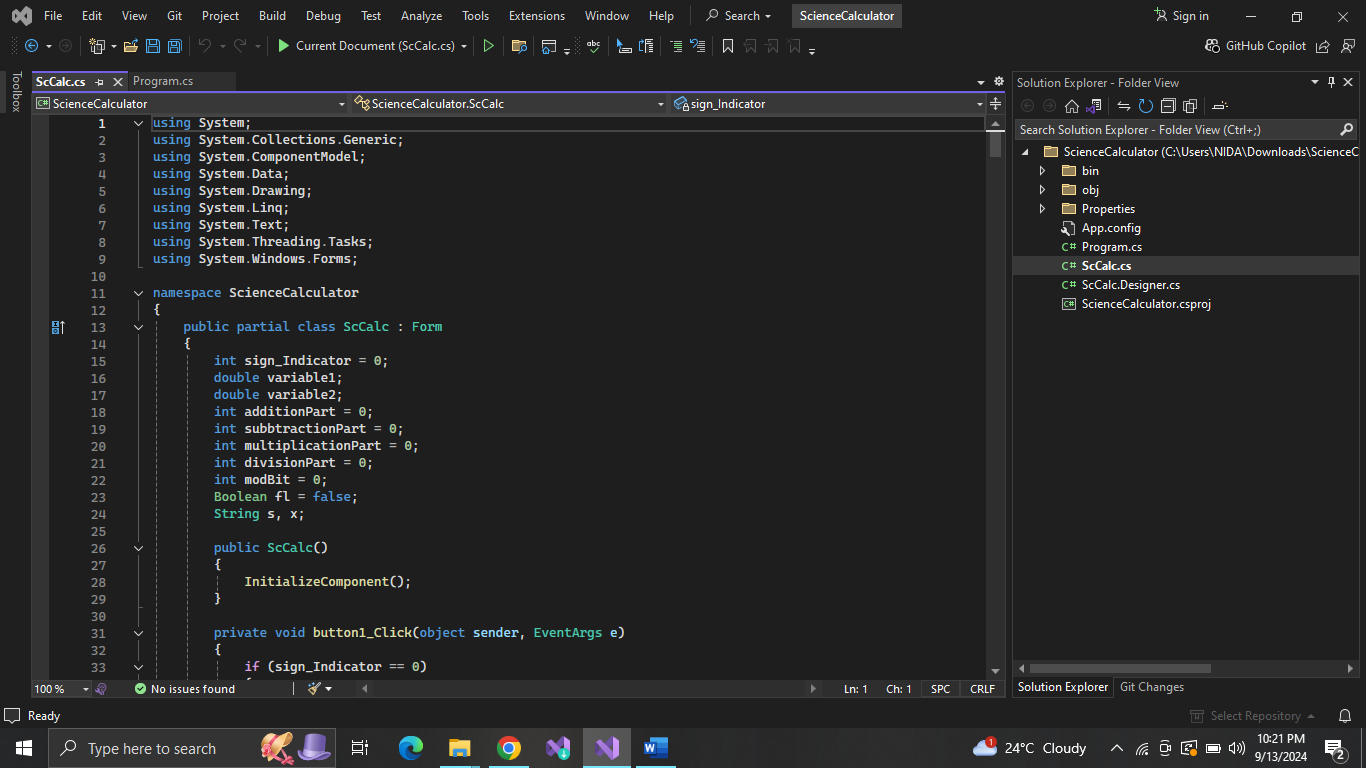
}

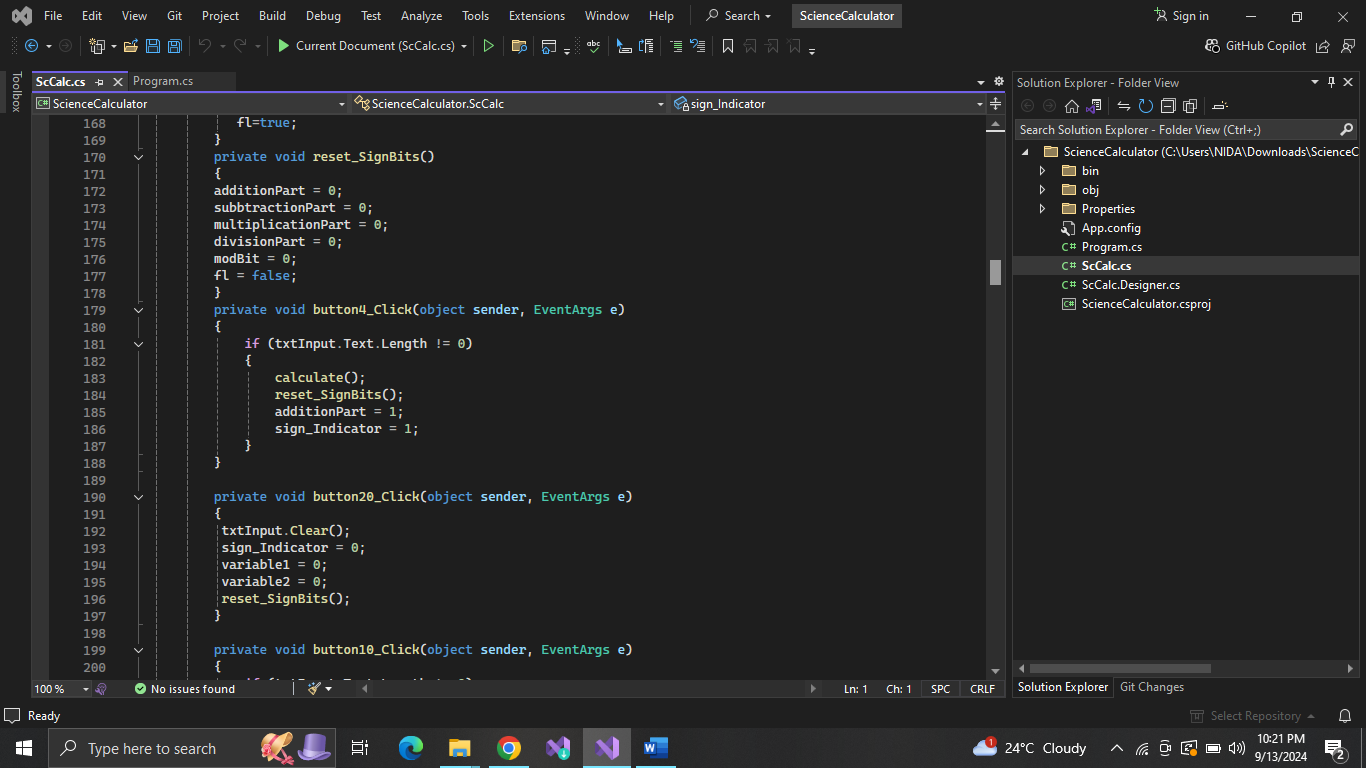
}

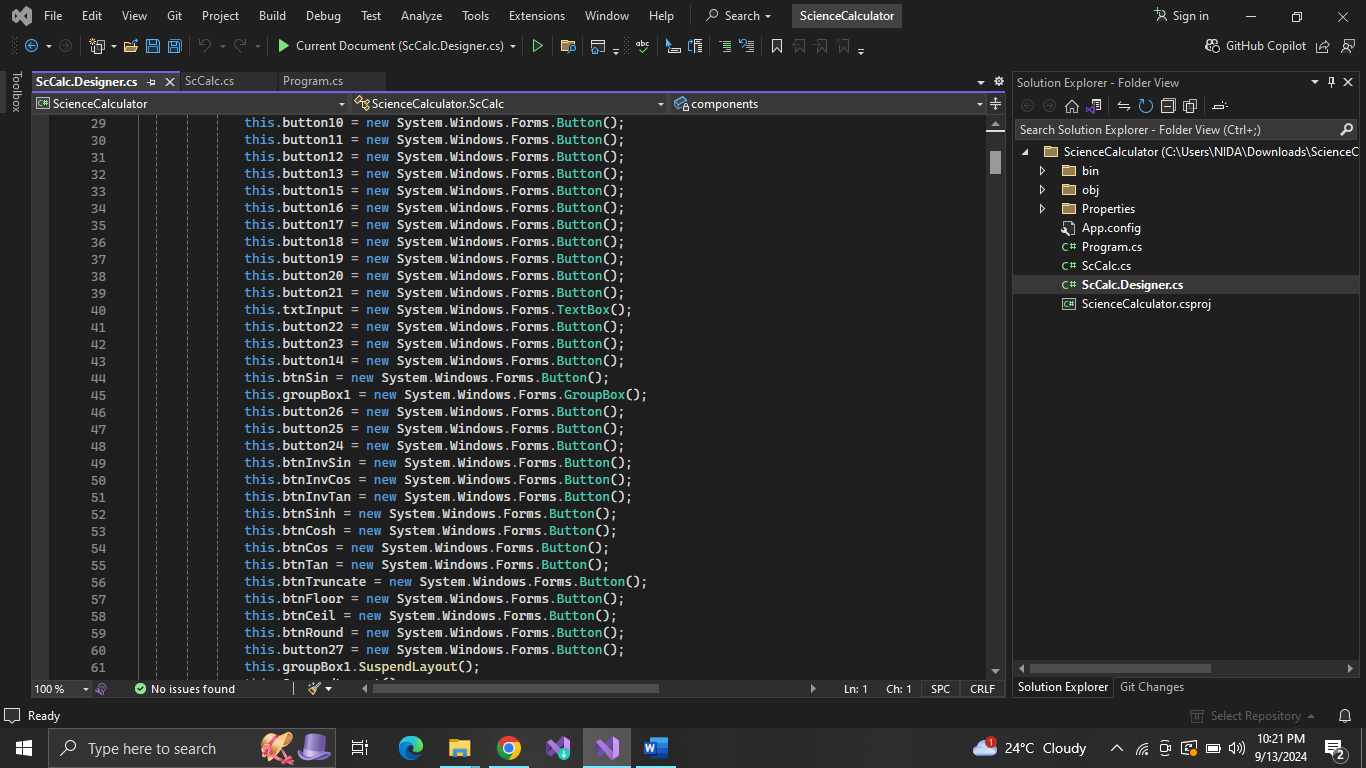
Graded Lab Tasks

LAB 01

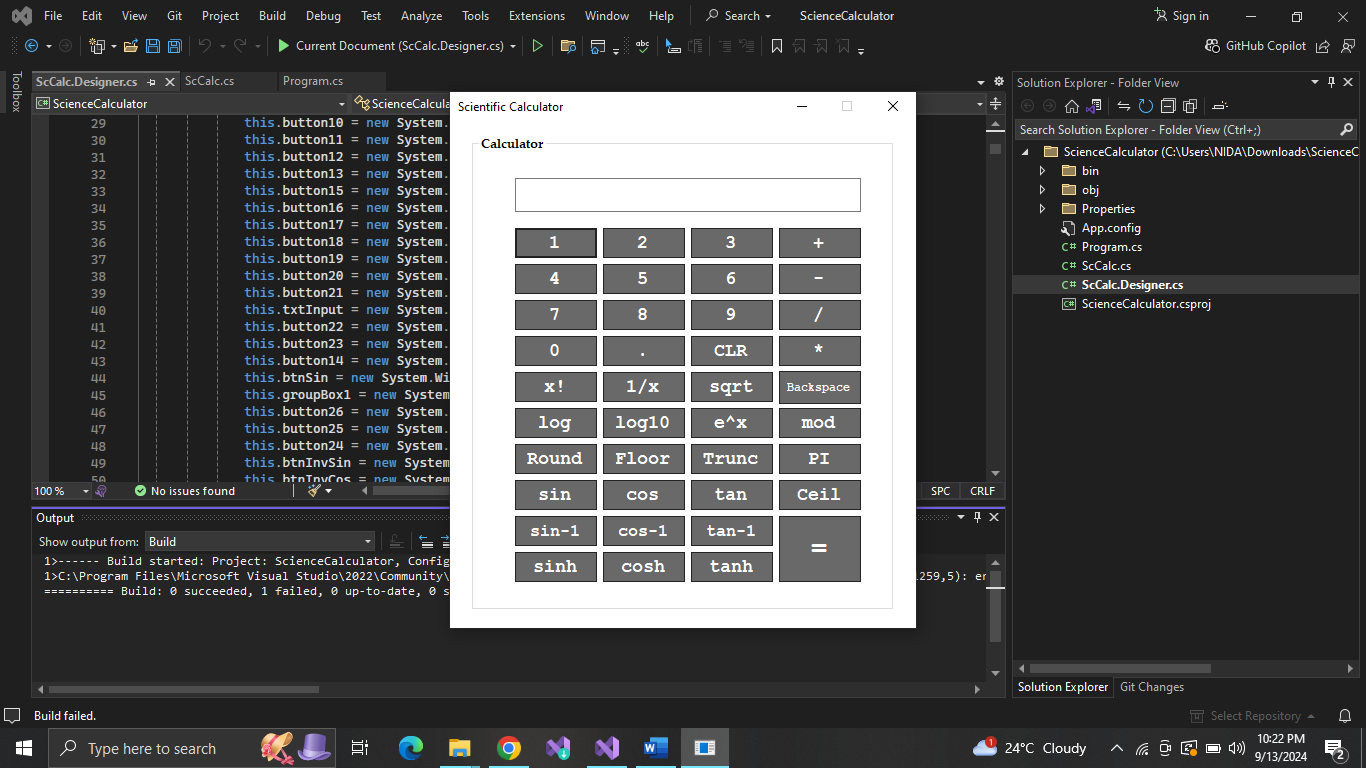
Task 1



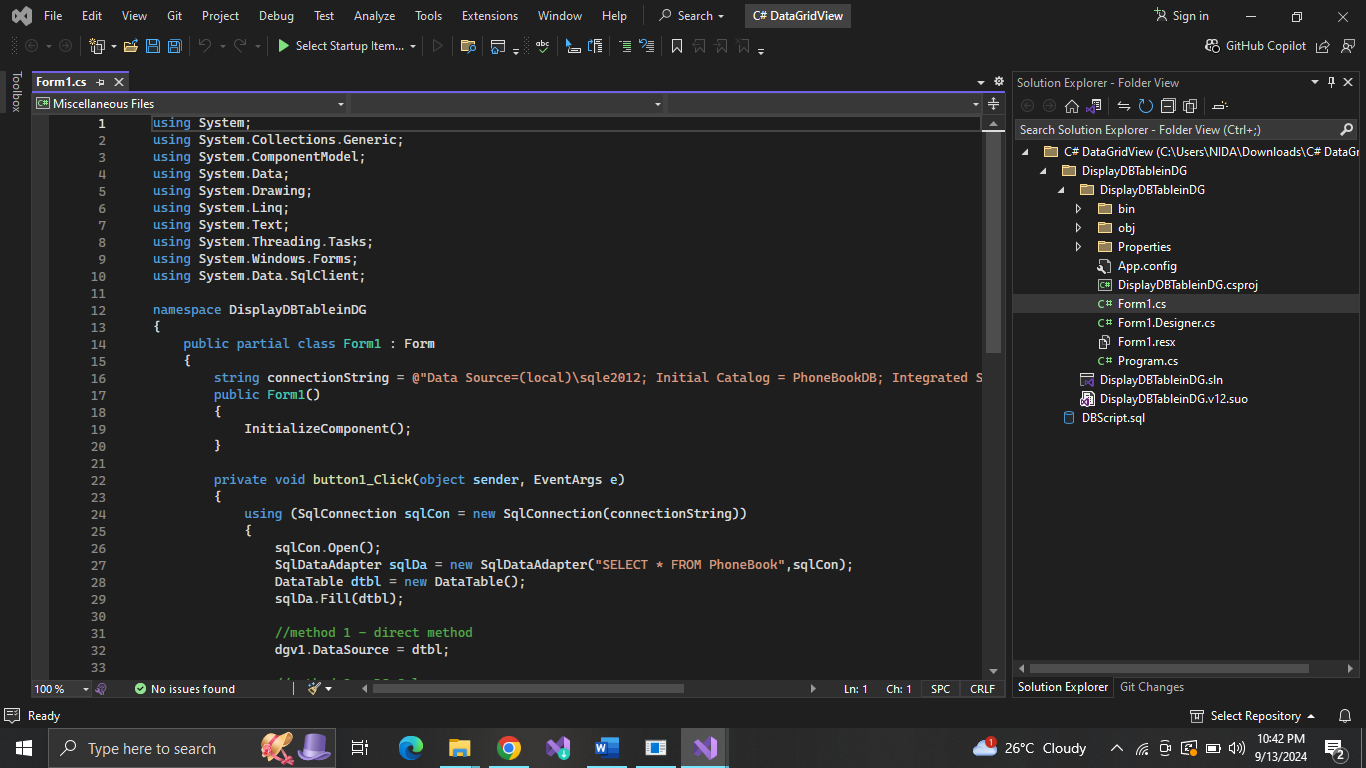


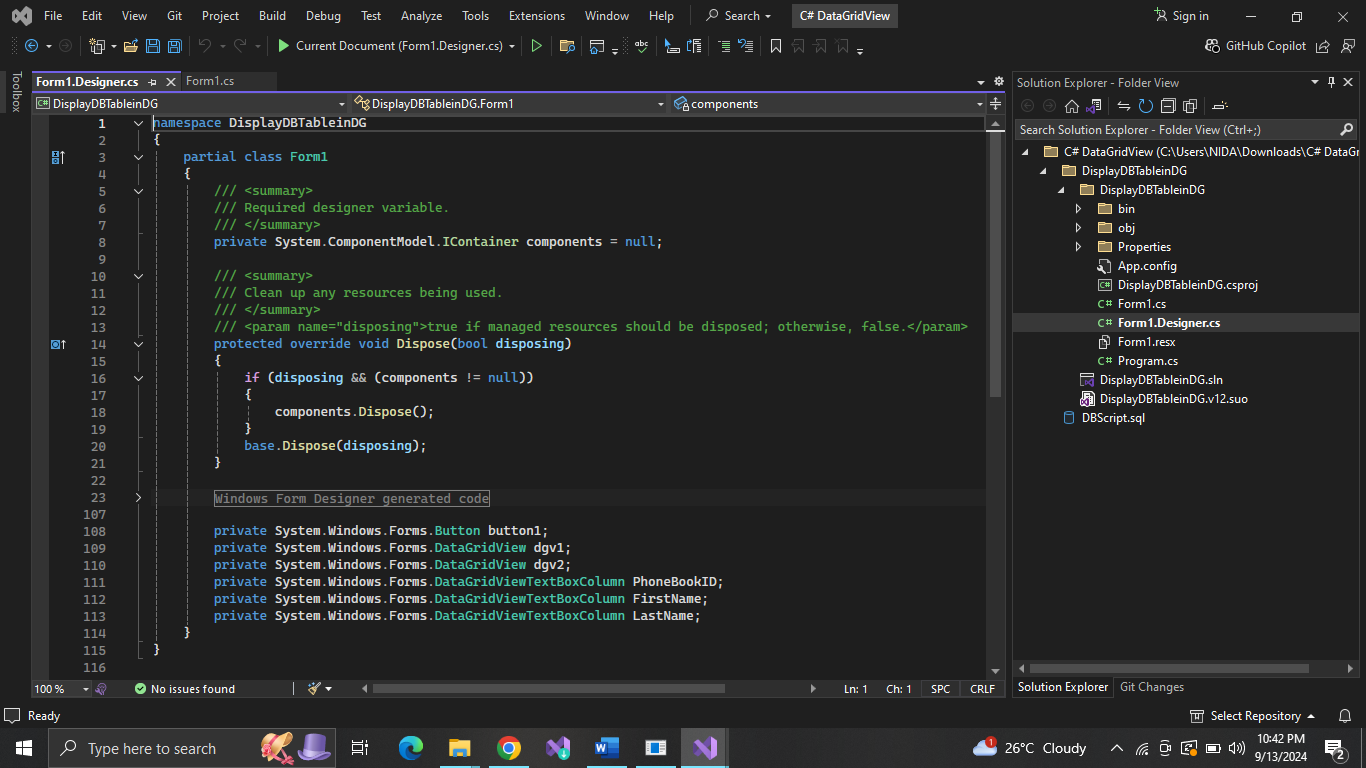


Ouput:

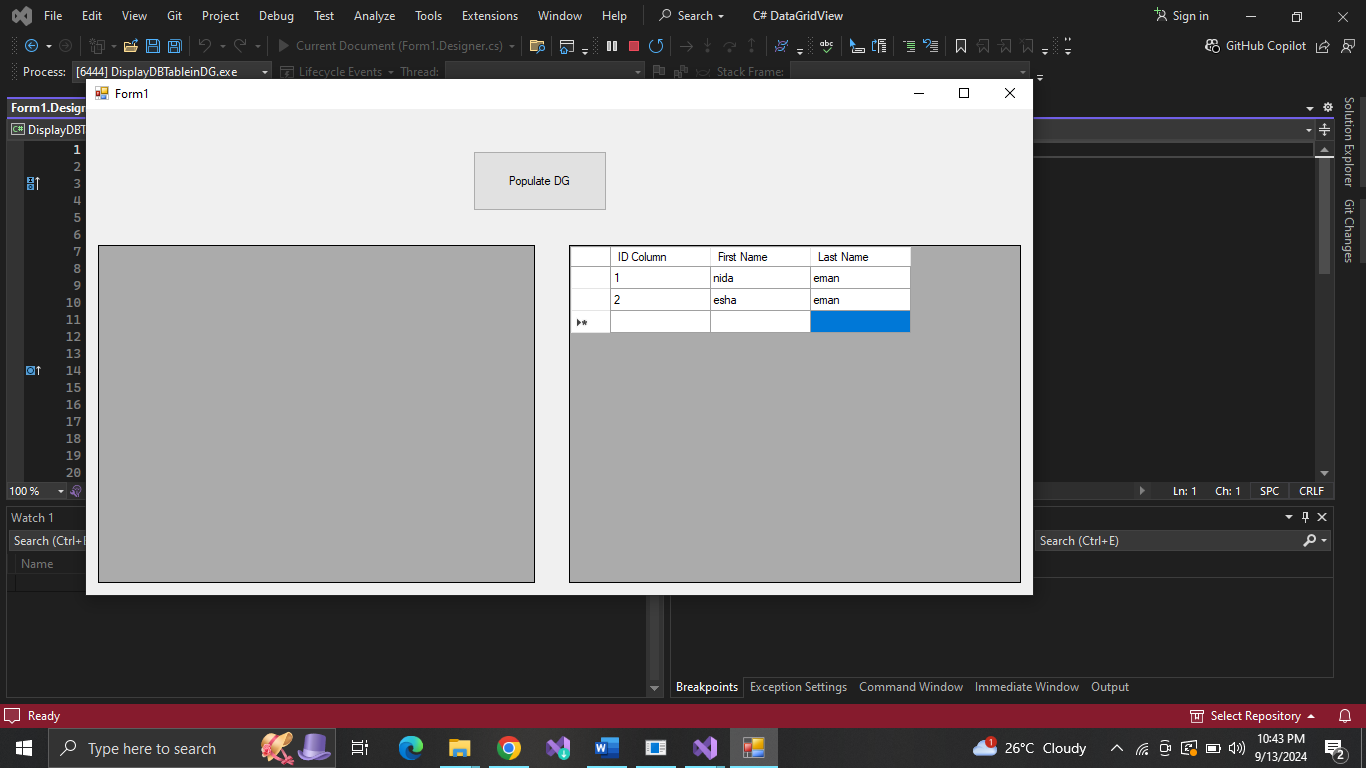


Task 02:



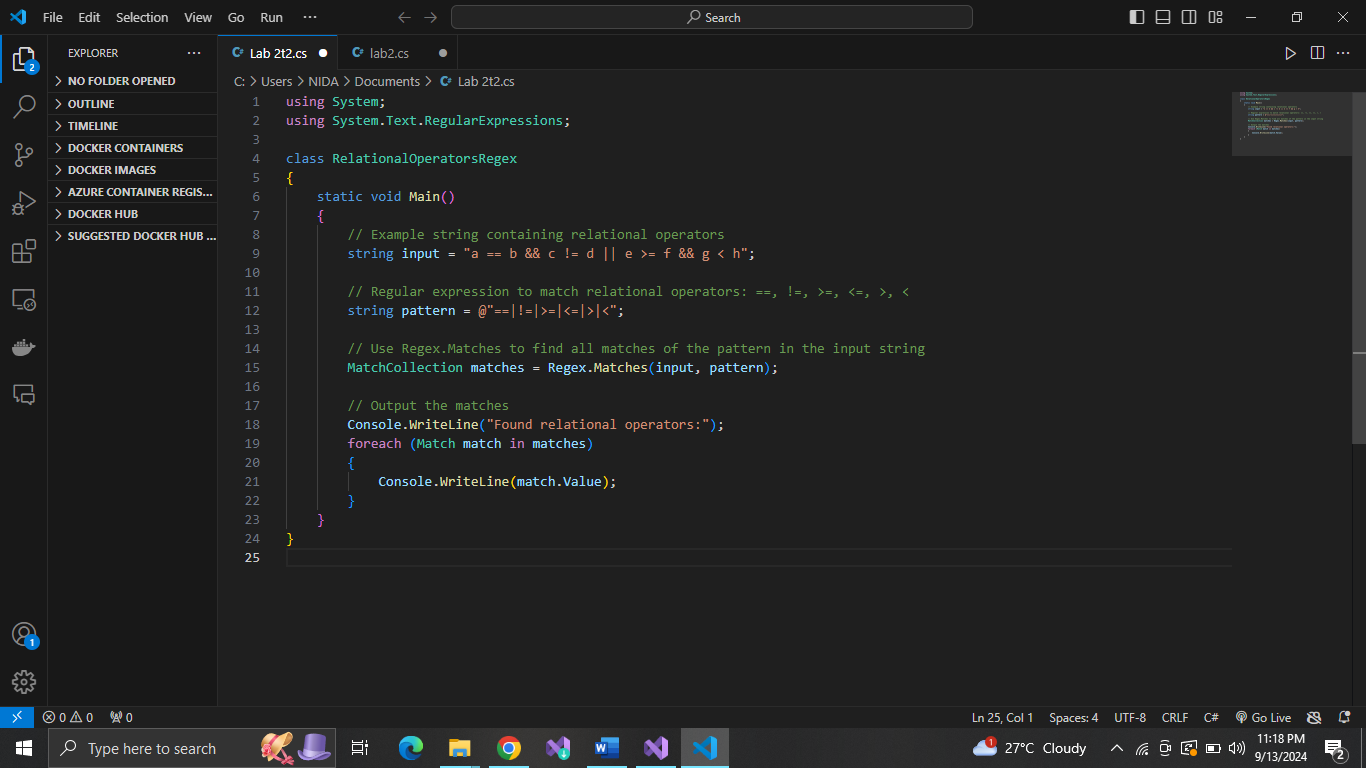


Output:



Lab 02 :

Task 02:



Task 01:

