**lab 3**

**Submitted By:**

Alveena ayesha

FA21-BCS-014

**Instructor:**

Mr. Syed Bilal Haider

**Course:**

CC

**Date:**

21st September, 2024

**Task#1:** Design a regular expression for floating point numbers having length not greater than 6.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab3Task1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

// Take input from a richtextbox/textbox

String input = richTextBox1.Text;

// Regular expression for floating point numbers with length not greater than 6

Regex regex = new Regex(@"^[-+]?[0-9]{1,5}(**\.**[0-9]{1,4})?$");

// Split the input on the basis of space

String[] numbers = input.Split(' ');

// Clear richTextBox2 before showing results

richTextBox2.Clear();

for (int i = 0; i < numbers.Length; i++)

{

Match match = regex.Match(numbers[i]);

if (match.Success)

{

richTextBox2.Text += numbers[i] + " "; // Display valid floating-point numbers

}

else

{

MessageBox.Show("Invalid number: " + numbers[i]);

}

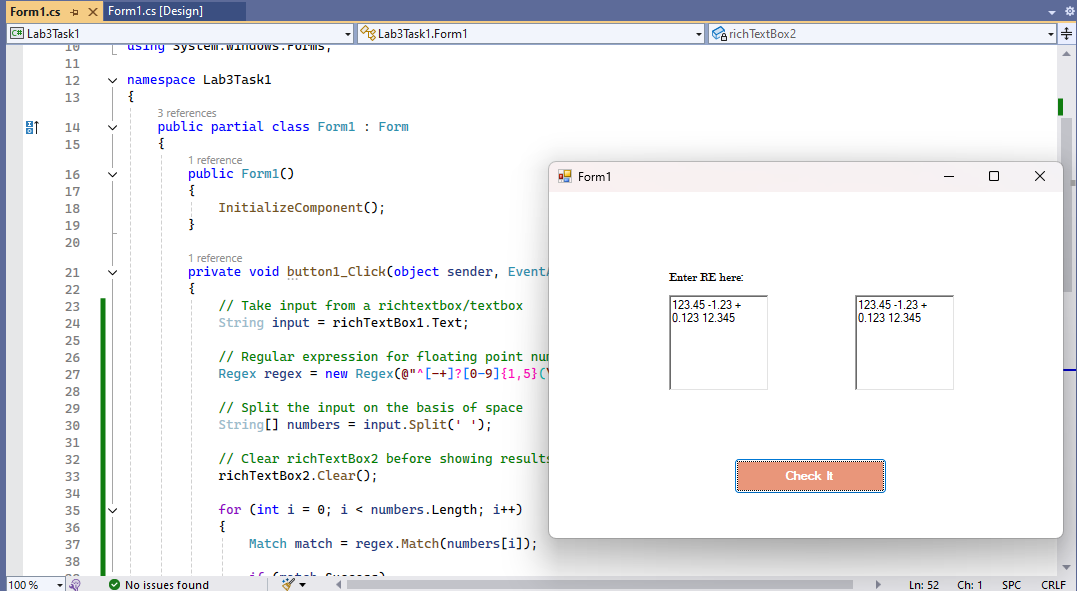
}

}

}

}

**Output:**



**Task#2:** Design a single regular expression for following numbers: 8e4, 5e-2 , 6e9 (Using Datagrid view)

.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab3Task2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

// Regular expression for scientific notation (e.g., 8e4, 5e-2, 6e9)

Regex regex = new Regex(@"^[0-9]+e[+-]?[0-9]+$");

// Loop through the DataGridView rows and validate each cell's value

foreach (DataGridViewRow row in dataGridView1.Rows)

{

// Skip the last row if it's empty (this is often the "new row" in DataGridViews)

if (!row.IsNewRow && row.Cells[0].Value != null)

{

// Get the input value from the first column (Number column)

string input = row.Cells[0].Value.ToString();

// Check if the input matches the scientific notation pattern

Match match = regex.Match(input);

if (match.Success)

{

// Add "Valid" to the second column (Status column)

row.Cells[1].Value = "Valid";

}

else

{

// Add "Invalid" to the second column (Status column)

row.Cells[1].Value = "Invalid";

}

}

}

}

}

}

**Output:**

