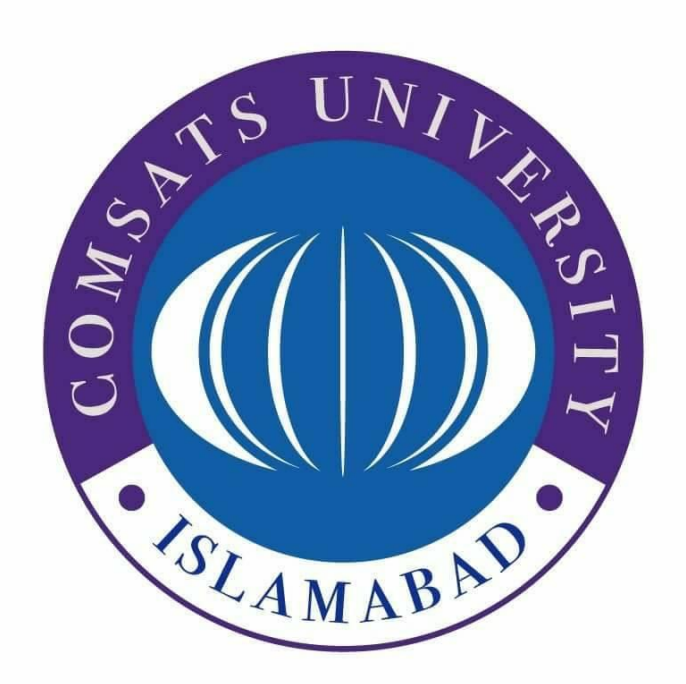


Group Members

Aiman Khatoon (Fa20-bcs-017)

Mahnoor (Fa20-bcs-045)

Submitted To: Sir Bilal Haider Bukhari

Date of Submission: 28-Dec-2023

Lab Terminal

Question 2:

2: functionalities along with screenshots (function code +output)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text.RegularExpressions;
using System.Windows.Forms;

namespace MathExpressionAnalyzer
{
   public partial class Form1 : Form
   {
      private string[] inputTokens;
      private int currentPosition;

      public Form1()
      {
         InitializeComponent();
      }

      private void analyzeButton_Click(object sender, EventArgs e)
      {
         string expression = textBox1.Text;
         inputTokens = Tokenize(expression);

         try
         {
```

```
        currentPosition = 0;
        AnalyzeExpression();
        label1.Text = "Analysis successful: Valid math expression!";
    }
    catch (Exception ex)
    {
        label1.Text = $"Error: {ex.Message}";
    }
}

private string[] Tokenize(string expression)
{
    return Regex.Split(expression, @"(\d+|[-+*/()])", RegexOptions.IgnorePatternWhitespace)
                .Where(s => !string.IsNullOrWhiteSpace(s)).ToArray();
}

private string GetCurrentToken()
{
    if (currentPosition < inputTokens.Length)
        return inputTokens[currentPosition];
    return "$"; // End of input marker
}

private void ConsumeToken()
{
    currentPosition++;
}

private void AnalyzeExpression()
{
```

```
    AnalyzeT();
    AnalyzeEPrime();
}

private void AnalyzeEPrime()
{
    string token = GetCurrentToken();
    if (token == "+")
    {
        ConsumeToken();
        AnalyzeT();
        AnalyzeEPrime();
    }
    else if (token == "$" || token == ")")
    {
        // E' -> ?
        // Do nothing
    }
    else
    {
        throw new Exception($"Unexpected token: {token}");
    }
}

private void AnalyzeT()
{
    AnalyzeF();
    AnalyzeTPrime();
}
```

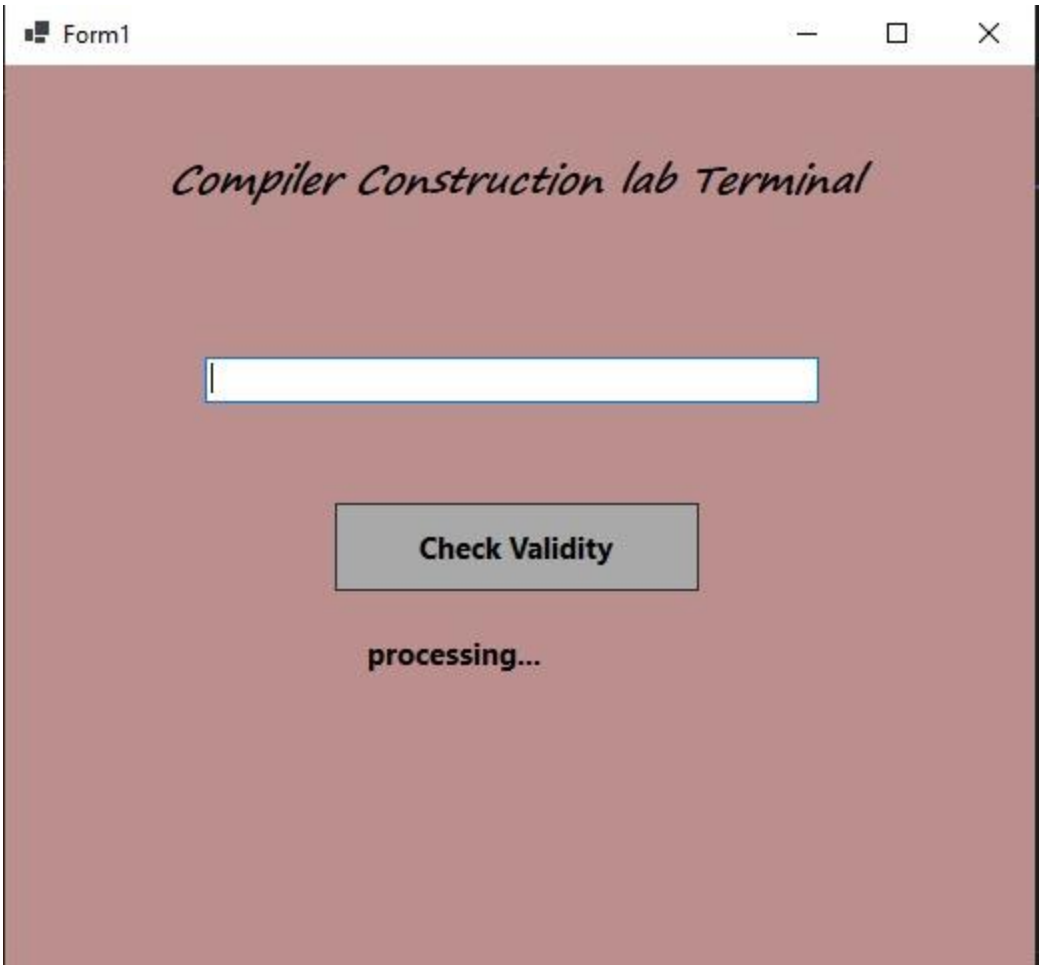
```
private void AnalyzeTPrime()
{
    string token = GetCurrentToken();
    if (token == "*")
    {
        ConsumeToken();
        AnalyzeF();
        AnalyzeTPrime();
    }
    else if (token == "+" || token == "$" || token == ")")
    {
        // T' -> ?
        // Do nothing
    }
    else
    {
        throw new Exception($"Unexpected token: {token}");
    }
}

private void AnalyzeF()
{
    string token = GetCurrentToken();
    if (token == "(")
    {
        ConsumeToken();
        AnalyzeExpression();
        if (GetCurrentToken() == ")")
        {
            ConsumeToken();
```

```
            }
            else
            {
                throw new Exception("Expected closing parenthesis");
            }
        }
        else if (int.TryParse(token, out _))
        {
            ConsumeToken();
        }
        else
        {
            throw new Exception($"Unexpected token: {token}");
        }
    }

    private void label2_Click(object sender, EventArgs e)
    {

    }
  }
}
```


Form1
Compiler Construction lab Terminal
Check Validity
processing...