Task 1:

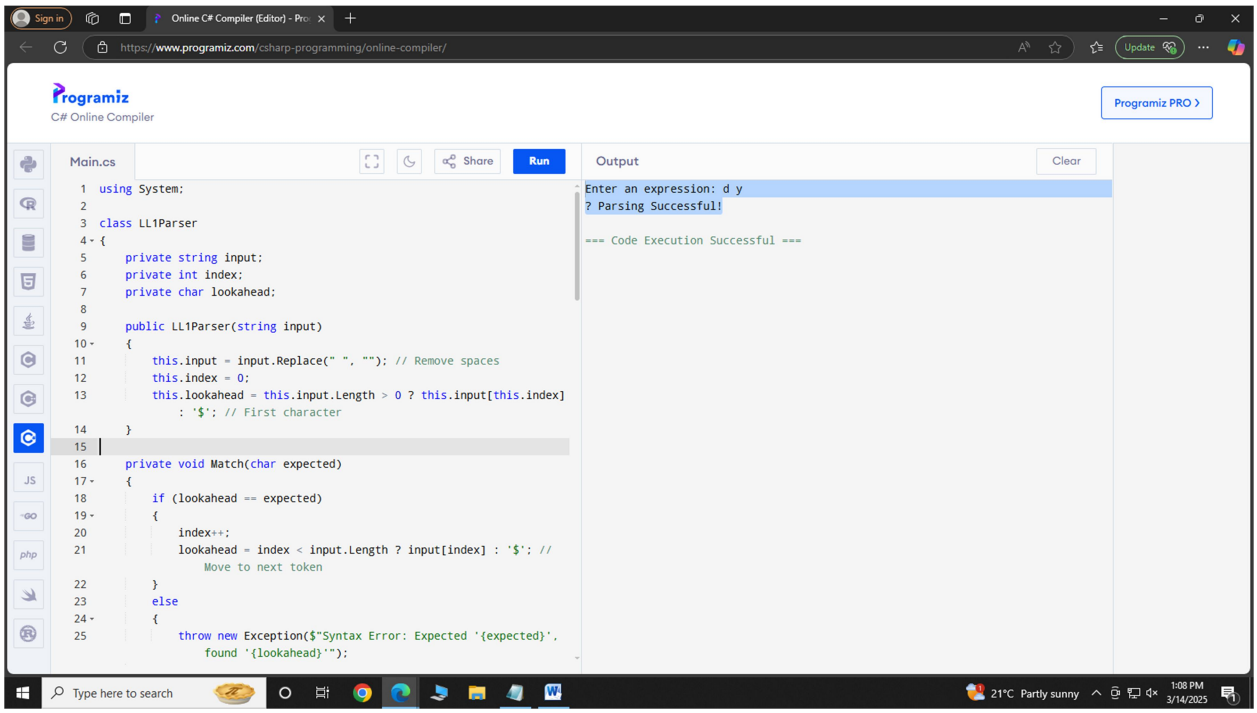
```
using System;


class LL1Parser
{
    private string input;
    private int index;
    private char lookahead;


    public LL1Parser(string input)
    {
        this.input = input.Replace(" ", ""); // Remove spaces
        this.index = 0;
        this.lookahead = this.input.Length > 0 ? this.input[this.index] : '$'; // First character
    }


    private void Match(char expected)
    {
        if (lookahead == expected)
        {
            index++;
            lookahead = index < input.Length ? input[index] : '$'; // Move to next token
        }
        else
        {
```

```
        throw new Exception($"Syntax Error: Expected '{expected}', found '{lookahead}'");
    }
}

public void Parse()
{
    S(); // Start parsing from 'S'
    if (lookahead == '$')
        Console.WriteLine("✅Parsing Successful!");
    else
        throw new Exception("❌Parsing Failed: Unexpected characters at end.");
}

private void S()
{
    if (lookahead == '(')
    {
        Match('(');
        C();
        Match('x');
        Match('y');
        SPrime();
    }
    if (lookahead == 'd')
    {
```

```
        Match('d');
        Match('y');
        SPrime();
    }
    if (lookahead == 'b')
    {
        Match('b');
        SPrime();
    }
    else
    {
        throw new Exception($"Syntax Error in S(): Unexpected '{lookahead}'");
    }
}

private void SPrime()
{
    if (lookahead == 'x')
    {
        Match('x');
        Match('y');
        SPrime();
    }
    // S' → ε (empty), do nothing
}
```

```
private void C()
{
    if (lookahead == 'e')
    {
        Match('e');
        CPrime();
    }
    else
    {
        throw new Exception($"Syntax Error in C(): Unexpected '{lookahead}'");
    }
}

private void CPrime()
{
    if (lookahead == 'm')
    {
        Match('m');
        CPrime();
    }
    // C' → ε (empty), do nothing
}

public static void Main()
```

```
    {
        Console.Write("Enter an expression: ");
        string input = Console.ReadLine();
        try
        {
            LL1Parser parser = new LL1Parser(input + "$"); // Append '$' as end marker
            parser.Parse();
        }
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
        }
    }
}
```



Task 2:

```
using System;


class LL1Parser
{
    private string input;
    private int index;
    private char lookahead;


    public LL1Parser(string input)
    {
        this.input = input.Replace(" ", ""); // Remove spaces
        this.index = 0;
        this.lookahead = this.input.Length > 0 ? this.input[this.index] : '$'; // First character
    }


    private void Match(char expected)
    {
        if (lookahead == expected)
        {
            index++;
            lookahead = index < input.Length ? input[index] : '$'; // Move to next token
        }
        else
```

```
    {
        throw new Exception($"Syntax Error: Expected '{expected}', found '{lookahead}'");
    }
}

public void Parse()
{
    S(); // Start parsing from 'S'
    if (lookahead == '$')
        Console.WriteLine("✅Parsing Successful!");
    else
        throw new Exception("❌Parsing Failed: Unexpected characters at end.");
}

private void S()
{
    if (lookahead == '(')
    {
        Match('(');
        C();
        Match('x');
        Match('y');
        SPrime();
    }
```

```
    else if (lookahead == 'd')
    {
        Match('d');
        Match('y');
        SPrime();
    }
    else if (lookahead == 'b')
    {
        Match('b');
        SPrime();
    }
    else
    {
        throw new Exception($"Syntax Error in S(): Unexpected '{lookahead}'");
    }
}

private void SPrime()
{
    if (lookahead == 'x')
    {
        Match('x');
        Match('y');
        SPrime();
```

```
    }
    // S' → ε (empty), do nothing
}

private void C()
{
    if (lookahead == 'e')
    {
        Match('e');
        CPrime();
    }
    else
    {
        throw new Exception($"Syntax Error in C(): Unexpected '{lookahead}'");
    }
}

private void CPrime()
{
    if (lookahead == 'm')
    {
        Match('m');
        CPrime();
    }
```

```
    // C' → ε (empty), do nothing
}
public static void Main()
{
    Console.Write("Enter an expression: ");
    string input = Console.ReadLine();
    try
    {
        LL1Parser parser = new LL1Parser(input + "$"); // Append '$' as end marker
        parser.Parse();
    }
    catch (Exception e {
        Console.WriteLine(e.Message);
    } }
```