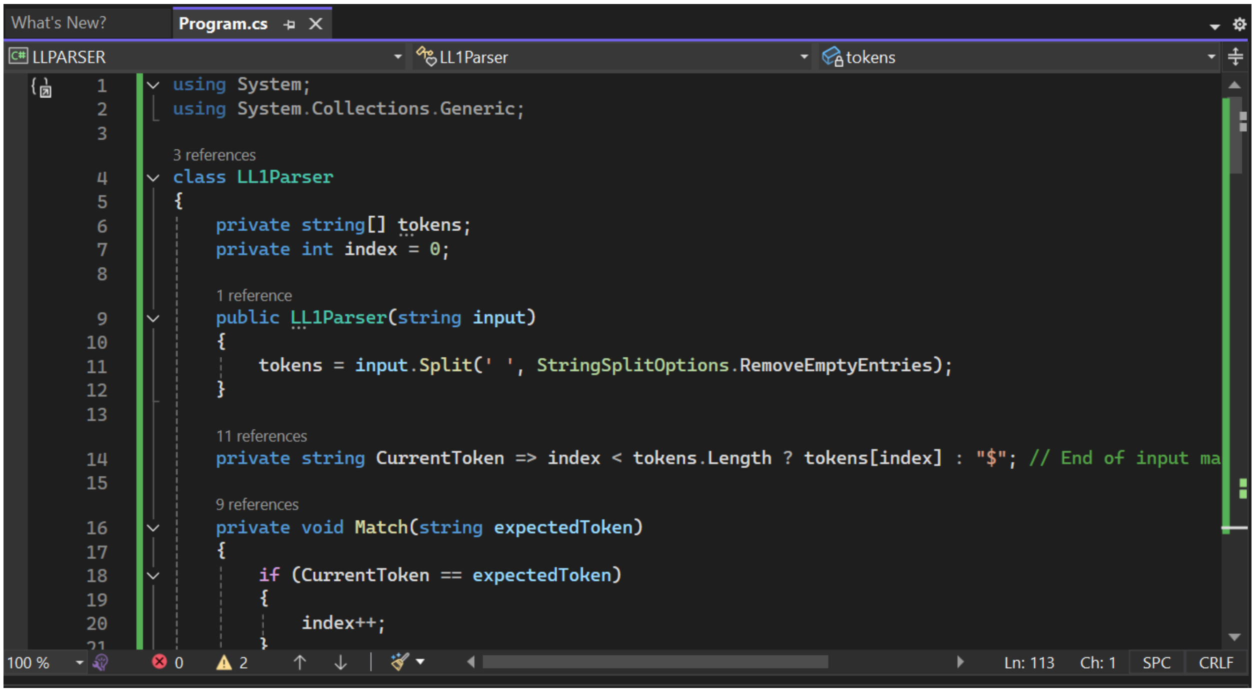
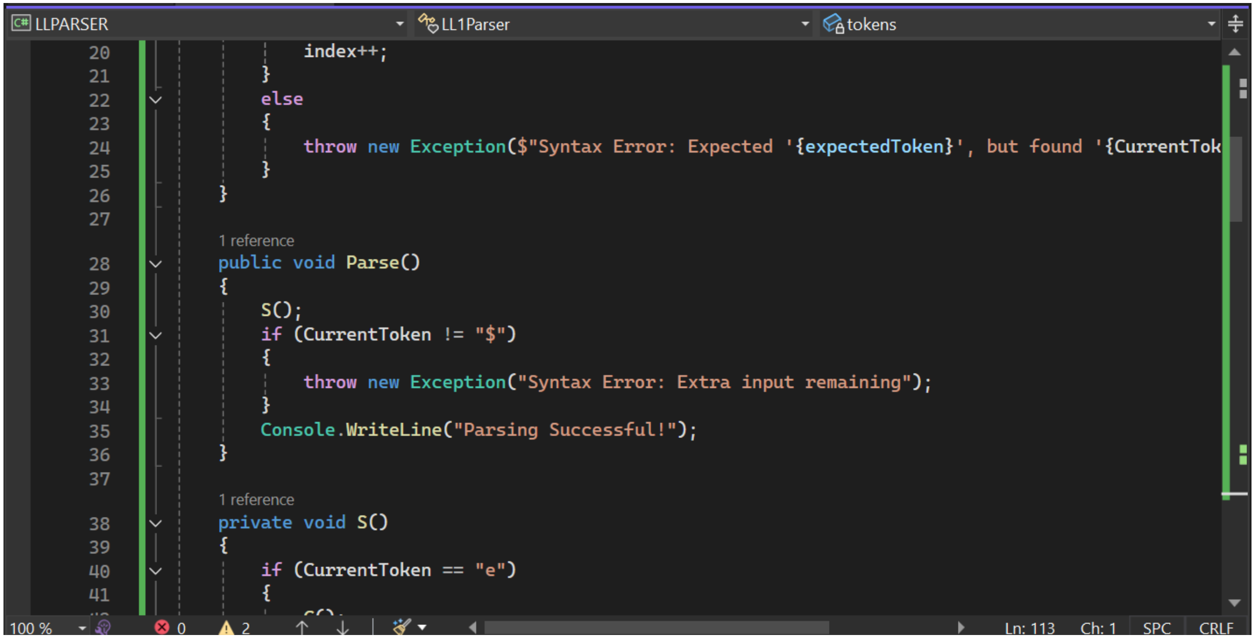
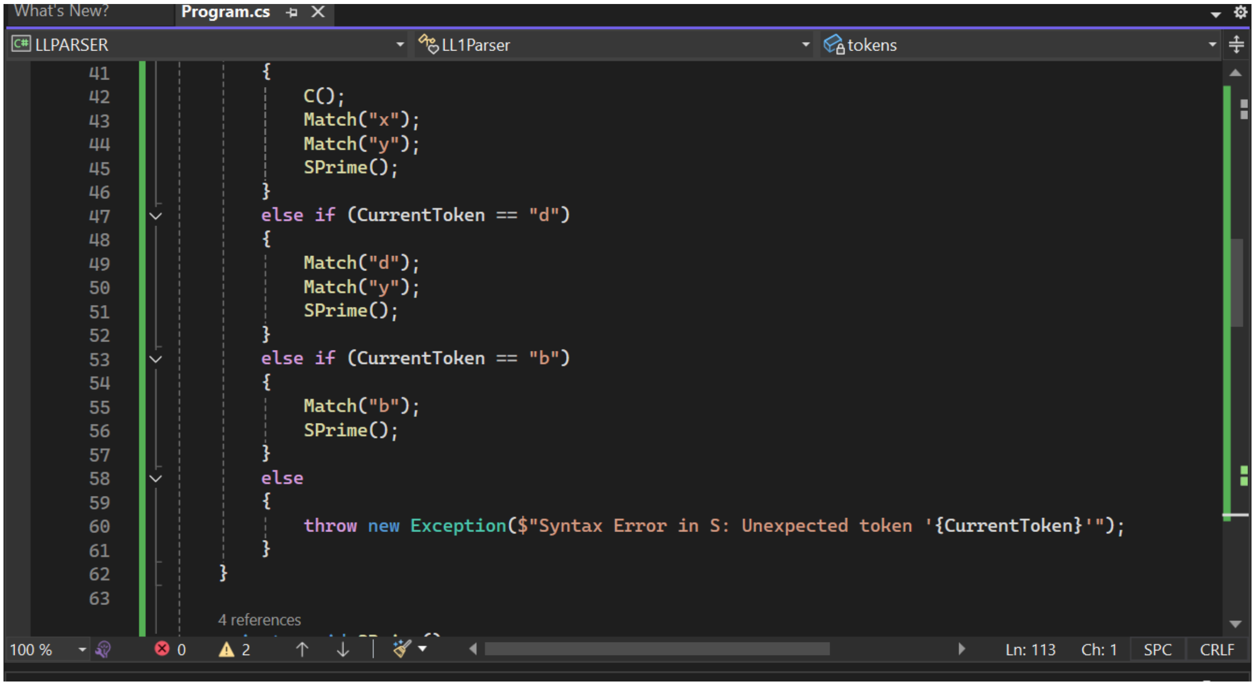


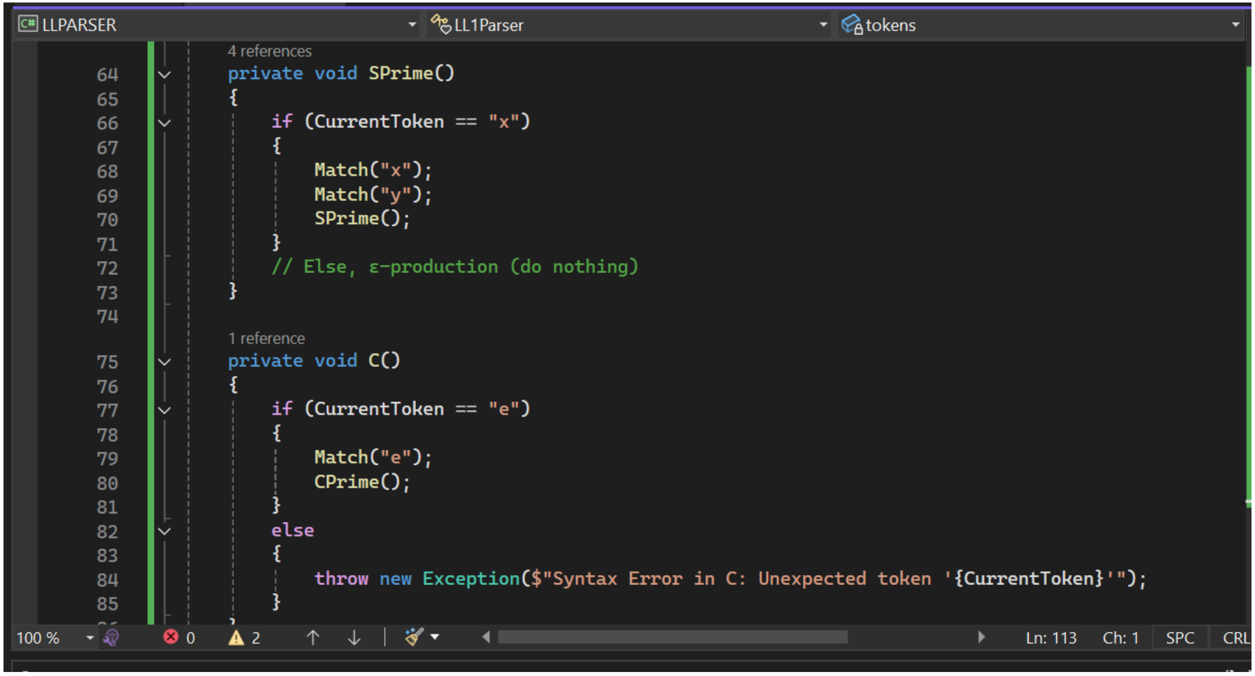
```csharp
using System;
using System.Collections.Generic;

class LL1Parser
{
    private string[] tokens;
    private int index = 0;

    public LL1Parser(string input)
    {
        tokens = input.Split(' ', StringSplitOptions.RemoveEmptyEntries);
    }

    private string CurrentToken => index < tokens.Length ? tokens[index] : "$"; // End of input ma

    private void Match(string expectedToken)
    {
        if (CurrentToken == expectedToken)
        {
            index++;
        }
```



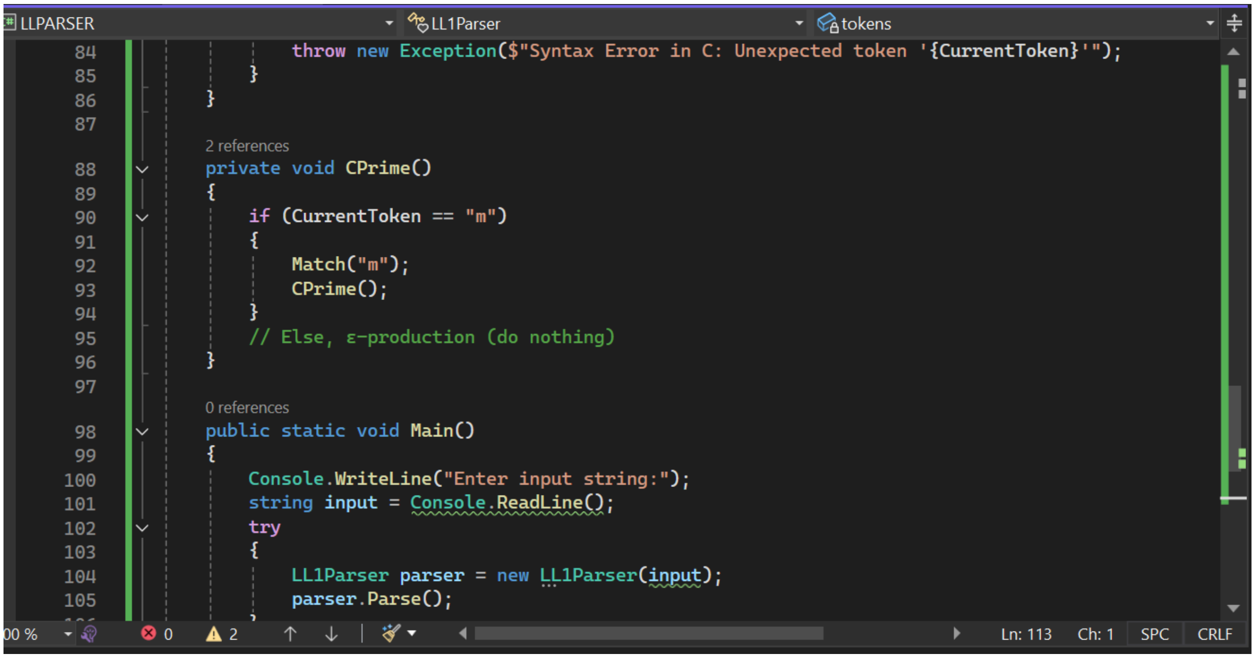
```csharp
            index++;
        }
        else
        {
            throw new Exception($"Syntax Error: Expected '{expectedToken}', but found '{CurrentTok
        }
    }

    public void Parse()
    {
        S();
        if (CurrentToken != "$")
        {
            throw new Exception("Syntax Error: Extra input remaining");
        }
        Console.WriteLine("Parsing Successful!");
    }

    private void S()
    {
        if (CurrentToken == "e")
        {
            S();
```



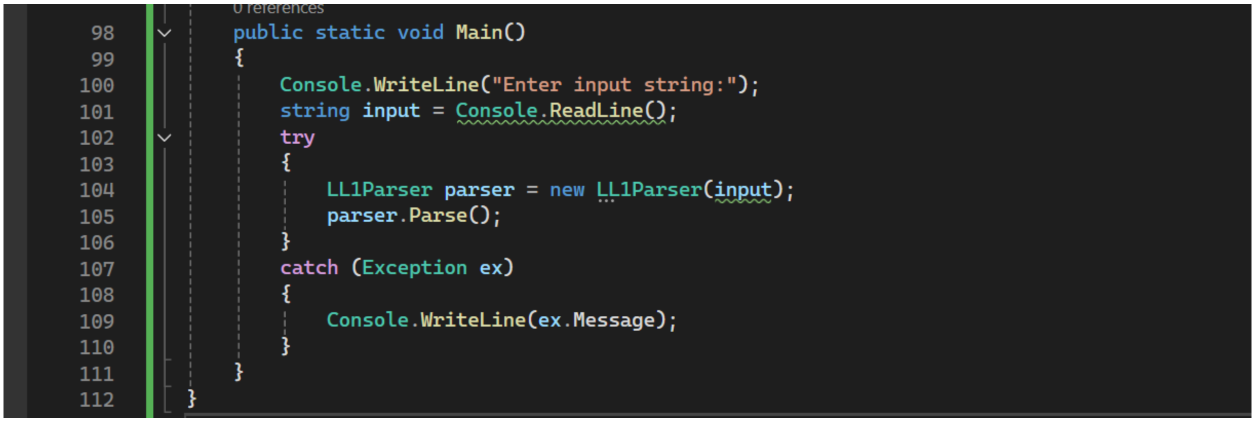
```
            {
                C();
                Match("x");
                Match("y");
                SPrime();
            }
            else if (CurrentToken == "d")
            {
                Match("d");
                Match("y");
                SPrime();
            }
            else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

```



```
        private void SPrime()
        {
            if (CurrentToken == "x")
            {
                Match("x");
                Match("y");
                SPrime();
            }
            // Else, ε-production (do nothing)
        }

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

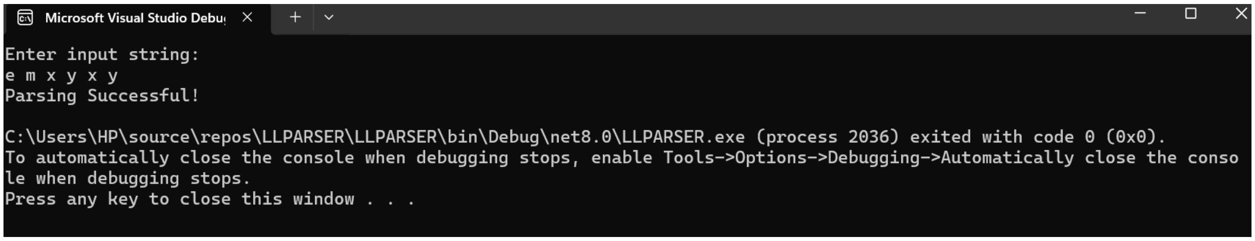


```
                throw new Exception($"Syntax Error in C: Unexpected token '{CurrentToken}'");
            }
        }

        2 references
        private void CPrime()
        {
            if (CurrentToken == "m")
            {
                Match("m");
                CPrime();
            }
            // Else, ε-production (do nothing)
        }

        0 references
        public static void Main()
        {
            Console.WriteLine("Enter input string:");
            string input = Console.ReadLine();
            try
            {
                LL1Parser parser = new LL1Parser(input);
                parser.Parse();
```



```
        0 references
        public static void Main()
        {
            Console.WriteLine("Enter input string:");
            string input = Console.ReadLine();
            try
            {
                LL1Parser parser = new LL1Parser(input);
                parser.Parse();
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
            }
        }
    }
```

OUTPUT:



```
Enter input string:
e m x y x y
Parsing Successful!

C:\Users\HP\source\repos\LLPARSER\LLPARSER\bin\Debug\net8.0\LLPARSER.exe (process 2036) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```