

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();

    }

    //  $S' \rightarrow \epsilon$  (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'  $\rightarrow \epsilon$  (empty), do nothing
}
```

```
public static void Main()
```

```

{

    Console.WriteLine("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);

    }

}
}

```

The screenshot displays the Programiz C# Online Compiler interface. The code editor on the left contains the following C# code:

```

1 using System;
2
3 class LL1Parser
4 {
5     private string input;
6     private int index;
7     private char lookahead;
8
9     public LL1Parser(string input)
10    {
11        this.input = input.Replace(" ", ""); // Remove spaces
12        this.index = 0;
13        this.lookahead = this.input.Length > 0 ? this.input[this.index]
14            : '$'; // First character
15    }
16
17    private void Match(char expected)
18    {
19        if (lookahead == expected)
20        {
21            index++;
22            lookahead = index < input.Length ? input[index] : '$'; //
23            // Move to next token
24        }
25        else
26        {
27            throw new Exception($"Syntax Error: Expected '{expected}',
28                found '{lookahead}'");
29        }
30    }
31
32    public void Parse()
33    {
34        Match(input[index]);
35    }
36 }

```

The output window on the right shows the following text:

```

Enter an expression: d y
? Parsing Successful!
=== Code Execution Successful ===

```

The bottom status bar indicates the system temperature is 21°C, it is partly sunny, and the date is 3/14/2023.

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();
    }
}
```



```
}  
  
//  $S' \rightarrow \epsilon$  (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);

        })
    }
}

```

The screenshot displays the Programiz C# Online Compiler interface. The code editor on the left contains the following C# code:

```

1 using System;
2
3 class LL1Parser
4 {
5     private string input;
6     private int index;
7     private char lookahead;
8
9     public LL1Parser(string input)
10    {
11        this.input = input.Replace(" ", "");
12        this.index = 0;
13        this.lookahead = this.input.Length > 0 ? this.input[this.index] : '$';
14    }
15
16    private void Match(char expected)
17    {
18        if (lookahead == expected)
19        {
20            index++;
21            lookahead = index < input.Length ? input[index] : '$';
22        }
23        else
24        {
25            throw new Exception($"Syntax Error: Expected '{expected}', found '{lookahead}'");
26        }
27    }
28 }

```

The right-hand side of the interface shows the output area with the message: "Parsing Successful!". Below this, it indicates "Code Execution Successful". The bottom of the image shows a Windows taskbar with various application icons and a system clock indicating 1:10 PM on 3/14/2023.