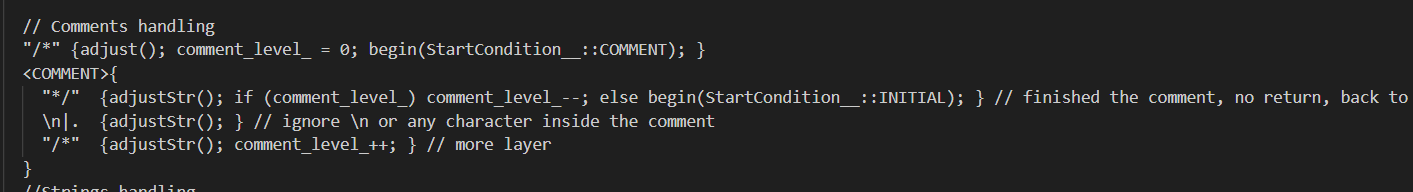
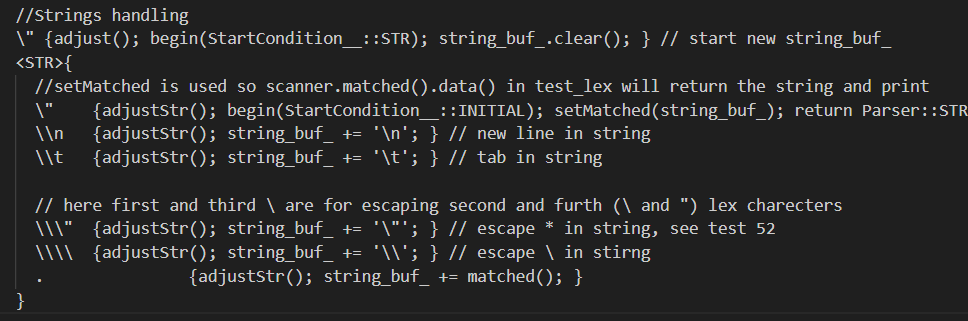
**How Comments Are Handled**

Comments are captured using the start condition COMMENT. When /\* is encountered, the lexer enters the COMMENT start condition and increments a comment nesting level (comment\_level\_). For every nested /\*, the nesting level is incremented. For every \*/, the nesting level is decremented. When the nesting level reaches zero, the lexer returns to the initial state.



**How Strings Are Handled**

String handling begins when a double-quote (") is encountered, transitioning to the STR state. During this state, newlines and tabs are represented as \n and \t and are directly added to the string buffer. Special characters like another double-quote or a backslash are escaped using double backslashes. The string is finalized when another double-quote is encountered, at which point the lexer transitions back to the initial state and returns the assembled string. *See interesting parts for 3 string edge cases.*



**Error handling**

Any character that doesn't fit the specified rules will trigger an "illegal token" error via errormsg\_->Error().



**End-of-file handling**

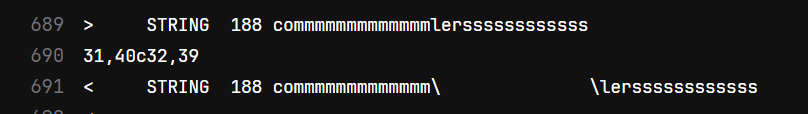
Simply return 0 (end the execution)



**Interesting parts**

3 edge cases in string: test52.tig

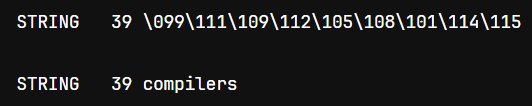
1.



Solution:



2.



Solution:



3.





Solution:



