Next moves with the lexer files:

A lexer is a component of a compiler that is responsible for breaking down the input source code into a sequence of tokens. C++ and Swift are two different programming languages, and they have their own lexical structure and rules. Therefore, a C++ lexer would not work directly in Swift without modification. So, our initial work from last semester will not work in the state it is now. However, it's possible to use a C++ lexer in a Swift project, but it would require us to create a wrapper or binding for the C++ lexer so that it can be called from Swift. This can be a complex process and would require knowledge of both languages and their interop capabilities.

This would kind of look like this process:

First define the C++ lexer interface: This involves defining the methods and data structures that the lexer uses to perform its operations. Second create a C++ wrapper: This involves creating a C++ wrapper class that provides a simplified interface to the lexer. The wrapper class should expose only the functionality that is needed by the Swift code, and should handle any data conversions or type translations that are required. Third create a C header file: This involves creating a C header file that defines the C++ wrapper class and its methods. This header file will be used by the Swift code to access the lexer functionality. Fourth import the C header file into the Swift code: This involves using the import statement to include the C header file in the Swift code. Fifth create Swift bindings: This involves creating Swift bindings for the C++ wrapper class methods, so that they can be called directly from the Swift code. Sixth test the wrapper and the binding: Once everything is set up, you can test your wrapper and the binding by calling the lexer functions from your swift code. This process would work however this process is very complex and can take a lot of our time. We can try to mitigate the time and complexity by using pre existing libraries and tools like boost.python, or SWIG.

This option seems more likely for us to use in the future of this project:

We will use a lexer that is written in pure swift or a library that is compatible with swift, this will avoid the additional complexity of wrapping or binding a C++ lexer. It's also worth noting that there are libraries available like antlr, flex, etc. These libraries are designed to generate lexer and parser for different languages, you can use them to generate a lexer for swift.