**Test cases**

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Actual Output** |
| test1.txt |  |  |

A black screen with blue text

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Actual Output** |
| test2.txt | Has 1 Lexical Error | Lexical Errors: 1 |

A computer screen shot of a black screen

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Actual Output** |
| test3.txt | Has all the added reserve words, lexemes, and tokens. | Compiled Successfully |

A computer screen shot of a black screen

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Actual Output** |
| test4.txt | Function with the reserved words compiled successfully. | Compiled successfully. |

A computer screen shot of a computer screen

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Actual Output** |
| test5.txt | Program that uses all the added operators like the /, -, <>, %, and ~ compiled successfully. | Compiled successfully. |

A screen shot of a computer

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Actual Output** |
| test6.txt | This tests if the lexemes I added or updated, like comment2, id, realliteral, hexadecimalinteger, and char all work. | Compiled successfully. |

A computer screen with text on it

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Actual Output** |
| test7.txt | Lexical Errors: 2 | Lexical Errors: 2 |

A screen shot of a computer

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **Input** | **Expected Output** | **Actual Output** |
| test8.txt | The underscores will be limited to 2 consecutives, and an identifier cannot start or end with them. There will be five lexical errors total.  The integers and hexadecimal integers work. The reals work, including the exponents. All operators and reserved words work. | Lexical errors: 5  Everything else is good.  Scroll below for screenshot. |

A screenshot of a computer program

Description automatically generated

**Approach**

I had a very hard time with this project because I did not understand C++. Adding and modifying all the lexemes and regular expressions was mostly easy. I just used what was already done as a guide. Some of the regular expressions took more time, such as id or adding the escape characters to char. Then came the part where we had to modify the functions in listing.cc. I had to basically learn C++ to understand what was happening line by line. I still don’t know where the appendError function even gets called. I don’t know where its arguments are being passed to it from. In the end, I just gave up and assumed that the arguments would come from “somewhere”. I added three static variables to keep a count of the types of errors. I changed the static string variable for the error message into a static queue. I had to learn how a queue works in C++. There was a lot of syntax that I just didn’t know because I’m not used to working with this language. But I completed all the changes and ran the test cases and I think everything worked, unless some of my expected outputs were actually supposed to catch some errors that I missed. In fact, the only thing it caught were Lexical Errors and no other types so I’m pretty sure some of those are wrong.

**Lessons Learned**

I learned how regular expressions are the rules that determine what characters a certain type of lexeme is allowed to be. I learned that after defining the lexemes, the scanner.l file then says what each of them will return if they are recognized. They might not return anything or they return a token, which is kept in another file. Finally I learned a bit about C++, which is a language I don’t really use.