ECE467 Lab 1 Summary

Name	Yannan Lin
Student Number	1001257891

This lab focuses on tokenizing the input stream of a MiniGLSL file. I declared the token types in parser.y then implemented the matching & tracing of tokens. This document describes some minor issue/challenge encountered when working on the lab.

Regex Ordering in Flex

In order to have tokens that are similar in nature recognized by lex properly, the ordering of regexes in the second session had to be carefully thought through. Although for the most part it doesn't matter, I had to come up with corner cases and avoid messing them up. One example is the standard floating point numbers (e.g. 1.23, 5.0) and the scientific notations of floating point numbers (e.g. 1.23e5). The scientific notation matching needs to come first before standard float matching. Otherwise we would get 2 tokens ["1.23", "e5"] and mishandle the intended semantics of the program.

Integer and Float Boundary Checks

To perform these range checks, simple atoi() and atof() are not good enough since they may or may not signal out of range conditions depending on platform implementation. After searching the internet I found some better solutions.

For integer, we first convert it to a long using strtol(text, 0, 10). Then compare it with the definitions of INT_MIN and INT_MAX from <limits.h> in the standard library. If it passes we cast the value back to int and report it to yylval.

For floating point numbers, strtof() gives a nice interface and error handling technique than atof(). If strtof(yytext) receives an OOB input, it sets the standard global variable "errno" to predefined error ERANGE. These helpers from <errno.h> enable clean detection for bad floating point numbers.