CS 152 Project Phase 1 Lab Report

Fall 2018 - Apollo Truong 861168740, Sidney Son 861214496

DEVELOPMENT

This is the first phase of the class project.

In this phase we used the *flex* tool to generate a lexical analyzer for the high-level source code language,“MINI-L”.

We used the in-class hand-out to define the following:

```
letter     [a-zA-Z]
digit      [0-9]
number     {digit}+
identifier {letter}({letter}|{digit}|[_]({letter}|{digit}))*
```

We then defined the tokens that will be read and set printf() statements to print the respective token for each symbol read.

Problems we ran into while developing the lexical analyzer were mostly syntax errors causing the output to print unwanted tokens. For example, we had misnamed the identifier by including an extra parenthesis which caused output to not recognize identifiers and output the identifier itself. Other syntax errors caused our program to have an overflow error and would not run our program.

USAGE

The usage can be simplified to the command:

```
$make
```

Information on Makefile

The file we wrote, mini_l.lex is followed by the command

```
$flex mini_l.lex
```

This generates the file

```
lex.yy.c
```

With this file, we then enter the following command

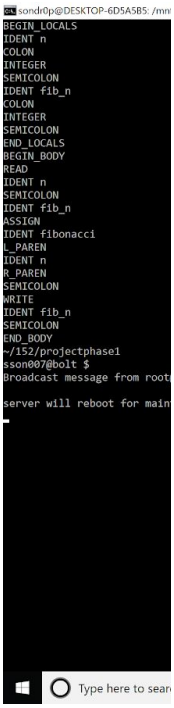
```
$gcc -o lexer lex.yy.c -lfl
```

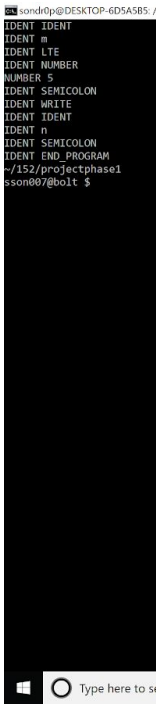
The generated “lexer” file is now able to convert the MINI-L (.min) program into a list of corresponding list of tokens.

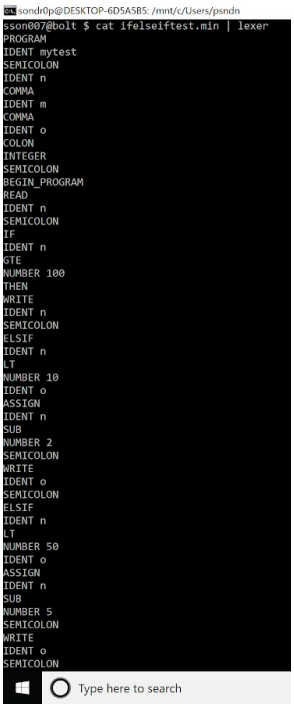
EXAMPLE

```
$cat fibonacci.min | lexer
```

Running the lexer on the files: `fibonacci.min` `dowhiletest.min` `ifelseiftest.min` will give the following output:







Running the diff command on in bash gave us a complete match of the respective `.token` files. Our Project Phase 1 is successful.