CSCE 3600: Systems Programming

Recitation Assignment 10 – Compiler Construction

Available: Week 13 Due: Week 14

On Linux, `flex` is the fast lexical analyzer generator. The file `scanner.lex` contains the description of tokens to generate a simple and very basic scanner using `flex`. To generate the scanner, use the following command:

```
flex scanner.lex
```

You should notice that `flex` created the file `lex.yy.c` in your current directory. Now compile this file into an executable program as follows:

```
gcc –o scanner lex.yy.c
```

Note that you must have the `zcalc.h` header file in the same directory for the compile to be successful.

The `scanner` will output the type of symbol that it recognized for the input that you typed. Now you can run the `scanner` executable and see what it does by typing in text and checking if it is recognized as a token. This scanner accepts a limited number of symbols such as "`(`", “`)`”, operator symbols (i.e., “`+`”, “`*`”, and “`–`“), numbers, and identifiers (i.e., variable names). Type in a few of these symbols and see how the `scanner` responds.

Now, let’s try to enter the division operator “`/`” and modulus operator “`%`”. What is the result? If the scanner does not recognize a symbol, it will simply not respond with the token, so it is vitally important that any valid symbol is accounted for.

You may use `Ctrl-D` to terminate the `scanner`.

Your task for this recitation assignment is to add `flex` support for the division and modulus operators in a similar fashion as is done for other arithmetic operators. To do this, you will need to modify the `scanner.lex` file to add support for these operator symbols in three locations:

- Add the constant definitions for the division and modulus operators. You may use any integer literal values following the existing set that are already defined.
- Add support for each symbol with the appropriate return value that matches the constant definitions added above.
- Inside `main`, add the appropriate `else if` branch and corresponding `printf` statement for each newly added symbol.

Now, run `flex` again and re-compile the newly created `lex.yy.c` file to make sure it works as expected.

REQUIREMENTS:

- No comments are required for this recitation assignment.
- Your program will be graded based largely on whether it works correctly on the CSE machines (e.g., `cse01`, `cse02`, …, `cse06`), so you should make sure that your program compiles and runs on a CSE machine.
- Although this assignment is to be submitted individually (i.e., each student will submit his/her own source code), you may receive assistance from your TA, IA (i.e., Grader), and even other classmates. Please remember that you are ultimately responsible for learning and comprehending this material as the recitation assignments are given in preparation for the future assignments and exams, which must be completed individually.

SUBMISSION:

- You will electronically submit your modified `scanner.lex` file to the **Recitation 10** dropbox in Canvas by the due date and time. No late recitation assignments will be accepted.