

CS 3323 - Principles of Programming Languages

Fall 2020 - Assignment 1

Download the files `pl-scanner.yy`, `driver.c`, `tokens.h` and `Makefile` from the assignment directory space. Then, perform the necessary modifications to the rule's file to recognize the below strings and tokens. The token codes can be found in the `tokens.h` header file.

The assignment is due on September 25th, 2020, 11:59pm CT. Files must be uploaded by then. Late policy deduction applies.

You can work in groups of up to 3 students. Groups remain the same for the whole semester. **All students must upload the corresponding files.**

Group members: At the top of your submitted flex file (the one with extension `.yy`), list the full names and student IDs (the 9 digit numbers) of your group members. List them above the `"#include "tokens.h"`, but under the line `"%{"`.

1. (1.5pt) Add the necessary rules to recognize the arithmetic operators: `+`, `-`, `*`, `/`, `<=`, `>=`, `==`, `!=`, `<`, `>`. See the file `tokens.h` to determine the constants to be used (they start with the prefix `'OP_'`).
2. (2pt) Modify the rule returning the `T_ID` token to recognize all identifiers matching the following:
 - identifiers can start with an underscore (`'_'`) or a letter; letters can only be lowercase.
 - all identifiers must have a minimum length of 3 characters; the potential `'_'` counts as 1 character of the minimum of 3.
 - identifiers can also consist of one or more digits, but cannot start with them.

Examples of valid identifiers are (commas used separate identifiers): `iter`, `iii`, `ali`, `_axe`, `b_p`, `b20`, `_counter_`, `_a_b_c_2_`, `_a3`

Examples of invalid identifiers are (commas used separate identifiers): `_`, `_2`, `at`, `2_`, `12b_5`

3. (2.5pt) Create a new rule to recognize floating point numbers. The rule should return the token `L_FLOAT`:
 - Can start with a digit, the `'+'` or a `'-'`
 - Both the integer and fractional part should consist of at least one digit
 - The integer and fractional part should be separated by a `'.'`

Examples of acceptable floating point numbers are: `"10.0"`, `"+1.5"`, `"-10.50000"`, `"0.9"`.

Examples of strings that should be rejected are: `"0."`, `".01"`

4. (1.5pt) Create a rule to recognize the following keywords: integer, float, foreach, begin, end, repeat, until, while, declare, if, then, print. The tokens corresponding to the keywords are those with the prefix 'K_' in the tokens.h header file.

For convenience, a Makefile is provided, but you are not required to use it. Run:

```
make
```

to rebuild the scanner generator (lex.yy.c), and to recompile the driver.

Several online resources can be found in the web, for instance:

- <http://alumni.cs.ucr.edu/~lgao/teaching/flex.html>,
- http://web.mit.edu/gnu/doc/html/flex_1.html,
- <https://westes.github.io/flex/manual/>.

More resources can be found by searching for the key terms: C scanner generator.

Do not change the driver file nor the tokens.h file since the actual integer values will be used for grading. Do not print anything to the output.

Every student should **upload a single file named: ABCDEFGHI.yy**, where ABCDEFGHI is the 9-digit code identifying the student (not the 4+4).