CS335 Assignment1

Dhruv Garg 210339

January 2024

1 Problem 1

Say the test file name is test.knp then run the commands in the order:

```
flex prob1.l
g++ -std=c++17 lex.yy.c -ll
./a.out test.knp
```

The output will be printed in the terminal.

Few Notes on special cases:

- Whenever there is an invalid string i.e. single quotes in double quotes or vice versa or there is an unclosed string then there is an error message printed in the terminal along with the lexemes which have been listed before the error is detected. For example: "abcd'abcd" will give an error.
- If there is a floating point like 0.123456789 then as told in Piazza, the scanner does the longest match and gives 0.123456 as float and 789 as integer as token.
- Similarly for 0x0123, it does the longest match of 0x0 and gives it a token of hexadecimal, and the number 123 is given the token integer.
- As given in Piazza, if there are keywords such as "BEGIN" and "Begin" then both these lexemes are given a count of 2.
- If there is an invalid character detected then the scanner prints all the lexemes listed till now and an error message showing the character along with the line number.
- If the test case has 0x123 and 0X123, I have maintained the same counter since nothing was mentioned. So if both are present exactly once then the output will print 2 for both.

2 Problem 2

Say the test file name is test.f08 then run the commands in order:

```
flex prob2.l
g++ -std=c++17 lex.yy.c -ll
./a.out test.f08
```

The output will be printed in the terminal.

Few Notes on special cases:

- As told in Piazza, I have removed the double quote (") from the special character and whenever there is an incomplete string or there is a single quote present in double quotes, it prints an error message along with all the lexemes listed till now. For example "abcd'abcd" gives an error message.
- As keywords, operators, Logical Literal, and names are case insensitive, I have increased the count for all of them if they are of the same type. For example, if the file has John and JOHN as names then output will have count for both of these as 2. This is the same as implemented in question 1.
- Whenever the error is detected, the scanner prints an error message and all the lexemes listed till now.