Project Phase1: Scanner for C- language

Due Date for Report and Code: September 10 Sunday, 2023, 10.00 PM.

Use LEX/Flex to implement a lexical analyzer for C language. Your lexical analyzer should support nested comments, and return appropriate error messages that are as meaningful as possible, such as comments and strings that don't end until the end of the file, etc. Your documentation should explain how you deal with comments, strings, errors etc.

- Study the grammar of the source language, and identify the lexical components of the language specifications.
- Study the input format for Flex and clearly understand how tokens are described.
- Identify the tokens to be returned by the scanner.
- Develop Flex script for the tokens identified above. Be careful with the definition of constants, the string literal and comments.
- Design symbol table. Use hash organisation for design.
- Design routines to produce the scanner output in the desired format as described in the Guidelines on how to present the results.
- Generate the scanner.

Guidelines on how to present the results

- When a token is identified, print its details . Thus your scanner should produce the list of tokens for the source program input to it.
- At the end of the program, neatly print the symbol table.
- Examples should include some programs containing lexical errors. The error should be properly identified and indicated.

At minimum, the project report should include:

- An overview explaining your code including info about what doesn't work (if anything).
- Listing of the code for your scanner.
- a list of the tokens your scanner recognizes and what they mean.
- a picture of DFA underlying your scanner. Be sure to indicate start and accepting states.
- a discussion of any assumptions you made beyond what is in the basic language description.
- A report on the test cases you ran along with the results(Include screenshots of output). Report any failures.

File name convention for the report: RollNo_Phase1.