1. Objective

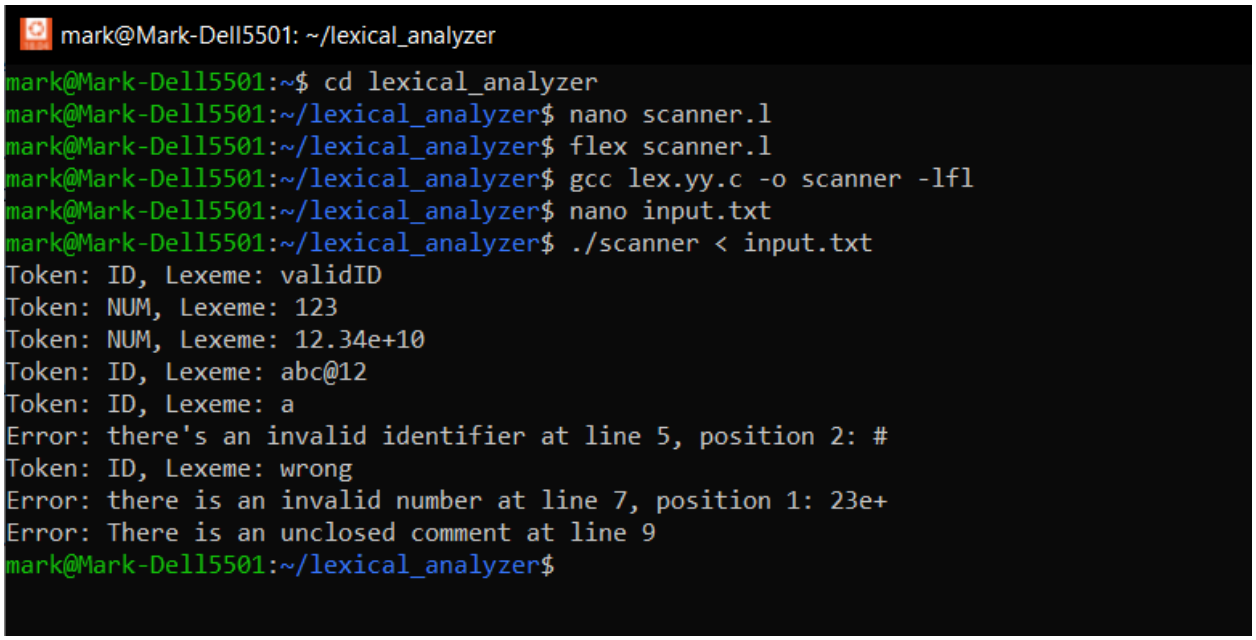
This project implements a lexical analyzer using Lex (Flex) for a simple programming language. The scanner identifies identifiers (ID), numbers (NUM), and comments, and handles several lexical errors.

2. Key Features & Decisions

- **ID Rule**: Starts with a letter, followed by letters/digits. It may include `@`, `$`, or `_` followed by digits.
- **NUM Rule**: Integer and float numbers, with optional scientific notation (e.g., `1.23e+4`).
- **Comments**: `/* ... */` style. If unclosed by EOF, an error is reported.
- **Error Handling**:
  - **Unclosed comments**
  - **Invalid identifiers** (e.g., a#)
  - **Invalid numbers** (e.g., 23e+)
  - **Unrecognized characters**

3. How to Run the Program

On the Ubuntu 18.04 Bash, these are the commands in the correct order to write, execute, compile, and test the program.



Mark Kyrollos - ID 900211436

Test File (input.txt):

validID
123
12.34e+10
abc@12
a#
wrong
23e+
/* unclosed comment starts here

Expected Output after compilation (from image):

Token: ID, Lexeme: validID Joken: NUM, Lexeme: 123
Token: NUM, Lexeme: 12.34e+10
Token: ID, Lexeme: abc@12
Token: ID, Lexeme: a
Error: there's an invalid identifier at line 5, position 2: #
Token: ID, Lexeme: wrong
Error: there is an invalid number at line 7, position 1: 23e+ Error: There is an unclosed comment at line 9