Lab Assignment 1- Lexical Analysis

In this assignment, you will work on implementing a lexical analyzer in your preferred programming language.

We will consider basic expressions, **understand what are the different tokens (and patterns for the tokens)** in the considered language/constructs.

Implement a lexical analyzer (using any programming language) for the considered tokens/patterns. Your program should take a statement as input and return the sequence of tokens as output.

Let us consider the example grammar for simple algebraic expressions:

expression → expression '+' term **|** expression '-' term **|** term

term → term '*' factor **|** term '/' factor **|** factor

factor → identifier **|** constant **|** '(' expression ')'

The terminals which will be the tokens, and their corresponding patterns here are;

- **identifiers**, described by the rule**: letter.(letter+digit)***
- **constants**, described by the rule: **digit.digit***
- **operators**: {+, -, *, /}
- "(", ")"

Implement a lexical analyzer using any programming language. Your program should take a algebraic expression as input and return the corresponding sequence of tokens as output.

- **Example Input**: `temp1+25*(bal2-53)`
- **Expected output**:

```
(id, temp1)
(+, opr)
(const, 25)
(*, opr)
( (, )
(bal2, id)
(-, opr)
(const, 53)
( ), )
```
