Function Overview（XIN GAO）

A four-function arithmetic expression calculator that supports parentheses, addition, subtraction, multiplication, division, and decimal numbers.

- **Tokenize**: Lexical analysis of the string expression
- **to_postfix**: Converts infix expressions to postfix notation
- **evaluate**: Uses a stack to evaluate the postfix expression
- **run_test**: Runs test cases to verify correctness

Modules Part：

1. read_number(line, index)

Parses integers and decimal numbers.
Returns a number token (e.g., `{'type': 'NUMBER', 'number': 3.14}`) and the updated index.

2. read_operator(line, index)

Parses operators and parentheses.
Returns a token such as `{'type': 'PLUS'}` or `{'type': 'LPAREN'}`.

3. tokenize(line)

Splits the input string into a list of tokens.

Input: `"1 + 2 * (3 - 4)"`

`Output:[{'type': 'NUMBER', 'number': 1}, {'type': 'PLUS'}, ...]`

4. to_postfix(tokens)

Converts an infix expression (`1 + 2 * 3`) into a postfix expression (`1 2 3 * +`), which is easier to evaluate.

5. evaluate(tokens)

Evaluates the postfix expression using a stack.
Returns the result as a float.

6. test(line) & run_test()

Compares the result from Python's built-in eval() with the result from the custom calculator, to verify correctness.

Test Coverage

- Basic operations: +, -, *, /
- Decimal calculations: 1.1 + 2.2
- Nested parentheses: ((1 + 2) * (3 - 4))
- Complex expressions: (1 + 2) * 3 - 4 / (5 - (6 - 7 * 8 + ...))