Subject: Formal Languages and Compilers
Name: Polina Sevastianova

Instruction on how to run

Project is implemented in Python 3 and tested on Merlin. To run the program, use the following commands:
cd Compiler
python3 src/Compiler1.py < tests/case1.txt
python3 src/Compiler1.py < tests/case2.txt
python3 src/Compiler1.py < tests/case3.txt
python3 src/Compiler1.py < tests/case4.txt

Also can be possible to run with Makefile
cd Compiler
make test

There are 4 cases for testing in tests/directory (case1, case2, case3, case4). The program reads expressions line by line from the input file and prints the compiled output or an error message.

Program details

Language: Python 3
Library: re - regular expressions, sys - reading from files
Main components:

- Lexical analyzer - tokenizes input text into tokens
- Syntax analyzer - builds Abstract Syntax Tree (AST)
- Semantic analyzer - validates AST against function rules
- Optimizer - simplifies AST
- Code Generator - generates final expression string with correct precedence and associativity

Supported functions: add, sub, mul, div, mod, pow, tern

Known limitations / bugs:

Only the listed functions are supported

Error messages are basic and may not cover all edge cases

Ternary operators are generated without parentheses

AI/External Code References:

Regular expression patterns were clarified with AI assistance.

All AI-assisted parts are marked in comments.

Testing

case1.txt - just normal valid expressions

case2.txt - expressions that demonstrate optimization

case3.txt - invalid expressions to test error handling

case4.txt - other cases