Assignment 2 part C Report

Name: Jessica Vu

Organization of Programming Languages

Changes for the Exception-based syntax error recovery:

1. **Create the syntax error exception class**:
    - Created a `Syntax_Error` exception class to handle syntax errors in the parser. This exception is raised whenever a syntax error occurs, allowing the parser to jump to an error-handling section.
2. **Update the `match()` and `error()` functions**:
    - Modified the 2 functions to throw a `Syntax_Error` exception if the current token does not match the expected token, triggering error recovery.
3. **Exception Handling in Parsing Functions**:
    - Wrapped key parsing functions (`stmt()`, `cond()`, `expr()`) in `try-catch` blocks to handle `Syntax_Error` exceptions.
    - When a syntax error occurs, the `catch` block attempts to recover by skipping tokens until it finds one that can resume parsing based on the **FIRST** or **FOLLOW** sets of the non-terminal.
4. **Error Recovery in `catch` Blocks**:
    - Implemented error recovery strategies within each `catch` block:
        - Used a loop to scan tokens until one from the **FIRST** or **FOLLOW** set of the current parsing function is found.
        - Resumed parsing from that point, enabling the parser to continue processing the input and detect further errors.
    - This method allows the parser to skip over invalid tokens and resume parsing, effectively handling multiple errors in a single run.

Notes:

- To build the project: `make` and then `./parse`
- To run test cases in the Makefile: `make test`
- To clean the project: `make clean`