Assignment 2 Documentation

At first after reviewing some notes and making sure I had a good understanding I then turned to DeepSeek as my AI tool for this project. The AI essentially helped me in getting started by explaining key ideas and guiding me through unfamiliar code and areas where I would get stuck. This process included:

- AI helped clarify how recursive descent parsing works and how each nonterminal in the grammar corresponds to a function in the code.
- Provided starting examples of functions needed for the parser

To start I asked AI to help me get started by giving it context and letting it break down the grammar into separate parsing functions and explain how they should interact. The initial structure gave me a solid starting point.

- The AI assisted in breaking down the grammar into functions that would each handle a specific part of the syntax.
- I structured the program to have a parse function as the entry point, calling other helper functions recursively.
- The parse tree was represented as a nested list, allowing easy traversal and debugging.
- The AI suggested using pattern matching and helper functions to handle different syntax rules cleanly.

Once I had a basic understanding, I asked the AI for guidance on structuring my functions. It helped outline functions for each nonterminal and explained how they should return parsed results or errors. It suggested using nested lists to represent the parse tree, which made traversal and debugging easier.

- I started with simple parsing functions that could recognize and validate individual tokens.
- The AI guided me on how to manage token streams effectively, making sure that functions consumed tokens correctly.
- I wrote functions for parsing expressions, terms, and operators based on the grammar.
- As I tested each function, the AI helped debug common errors, such as incorrect recursion and improper token consumption.

- One of the biggest challenges was implementing meaningful error messages.
- The AI suggested tracking the line number and returning informative messages when syntax errors were found.

- Initially, my error messages were too vague, but iterating with AI's suggestions helped refine them.

Most of the AI use was to refine and debug my code. I ran into a lot of small syntax issues and logical errors, and the AI helped me work through them one by one. When I couldn't figure out the error messages, I copied my code into the chat, and the AI walked me through possible fixes. Some drawbacks included subtle issues, like misplaced parentheses or incorrect recursion handling. Other times it struggled to understand why my code wasn't working or it would suggest for me to edit my code into the same code I already had.

- I ran various test cases using provided input files and additional edge cases I came up with.
- The AI helped identify logic errors by analyzing my output compared to expected results.
- Some bugs were tricky to catch, such as incorrect handling of nested expressions or missing base cases in recursive functions.
- I learned to step through my parser function-by-function to isolate issues.

This leads to limitations I found. Sometimes, the AI didn't catch certain mistakes, especially when the logic error wasn't obvious from a single function. Most of the time It would have to be reminded of previous code or I would have to resend my code for analyses.

Overall working with this AI was a back-and-forth process, and I learned a few things about how to better take advantage and prompt this AI. In a way it's important to consider that you are walking the AI through the process rather than the other way around. It is important to have a fundamental understanding of the concepts used for this assignment to better assist the Ais understanding to help you better. That being said, It is important to give the AI as much context as possible to avoid diverging from the actual requirements. AI tends to overdo or under deliver when a broad prompt is given.

Furthermore, it is important to debug in smaller steps after feeding it your current code. Feeding smaller snippets of code helps get more precise debugging assistance while your entire code gives it context. The smaller snippets then allow you to walk AI step by step through your code to help you better find any bugs and errors. That being said, AI can also give a lot of incorrect responses or code that needs to be checked. There were several times where it would take me back and forth from one code to the next where both code pieces were wrong.

Some solutions I found helpful were to have two separate chats. The first chat was to get basic understanding, structure, and guidance when I would get stuck, while the other chat was used to debug with greater depth. This would allow me to be more specific with the chat that would help me debug.