Anh Tran
pat200000

Project 1

Approach:

- The approach that I took for the project is to break it off into 2 main parts, the basic prefix expression calculator, and the history list to reference.

Organization:

- I broke the prefix expression calculator into smaller parts - or helper functions that would help the main evaluate function, as this would help make debugging easier and don't overcomplicate the program.

Problems Encountered:

- The main problem I had with this part was the parsing logic as it was not parsing the string correctly so it couldn't evaluate the expression.
- Initially had issues with tokenizing input and managing whitespace.
- Figuring out how to manage error handling for invalid expression

Solutions:

- Change the type of the functions and the logic of the parser
- For whitespace in the expression, make the parser ignore them and continue on
- Any expressions that are not the ones we're considerinng automatically is an invalid expression

Learning:

- Gained a deeper understanding of prefix notation and recursive evaluation.
- Learned about the concepts of lazy evaluation of Haskell and how that affects the structure of programming in the language
- Learned about the concept of pure function and pure code and why we would need to do it in Haskell
- Learnt about efficient error handling and string manipulation in Haskell.

Incomplete Features:

- None. All features were implemented as per the specification.

Challenges:

- Parsing expressions correctly and maintaining a history were challenging but enlightening.
- Figuring out a way to add the evaluated expression into a list without adding complexity to the program