Idea 1.
Implementing operational semantics with the user can define Hoare logic assertions within the language (Pre-condition, Post-condition, loop invariants).
The system can verify the language satisfy the pre-condition and post-conditions. I plan to have Assign, while, if, Seq, Assert, Skip, and Invariant commands. I will define these commands' small-step semantics and the Hoare logic system in mathematical formal definition in my report.
Users can write programs with these commands and verify the program satisfy the pre and post-conditions before running the program.

Idea 2.
Implementing a dependent type system. The type system can verify(type check) a term satisfies its dependent type.

I plan to add Inductive type, dependent function type, sum type and product type. If there is still time, I will also add proposition type and make it a proof assistant and implement a goal printer that gives information to users on what needs to be proven and what is known now. Again, I will define these types as a type system in formal ways with horizontal lines.