

Projeto de Compiladores

Parte 1

O projeto de compiladores consistirá em ter um compilador funcional de uma versão reduzida da linguagem C até o fim da cadeira.

O projeto será dividido em entregas progressivas, onde vamos cobrar em cada uma a implementação de uma parte do compilador. É importante que não deixem de fazer as entregas, mesmo que não consigam enviar à tempo, pois a entrega seguinte dependerá da implementação correta do que foi requisitado anteriormente.

- **Especificação da Linguagem:**

Vocês precisarão definir uma gramática que representa um subconjunto da linguagem C. Esta gramática precisará ter as seguintes regras:

- **file:** regra raiz da gramática.
- **function_definition:** regra que descreve a definição de uma função.
- **arguments:** regra que define o formato dos argumentos na declaração de uma função.
- **body:** regra que descreve um statement dentro de chaves ('{' e '}').
- **statement:** regra que conterá diferentes tipos de statements.
- **if_statement:** regra que define o formato de um 'if'.
- **else_statement:** regra que define o formato de um 'else'.
- **for_loop:** regra que define o formato do 'for'.
- **for_initializer:** regra que define o formato do primeiro parâmetro do 'for', o inicializador.
- **for_condition:** regra que define o formato da condição do 'for'.
- **for_step:** regra que define o formato do *step* do 'for'.
- **variable_definition:** regra que define o formato de uma definição de variável.

- **variable_assignment:** regra que define o formato da atribuição de um valor a uma variável.
- **expression:** regra que define os diferentes tipos de expressões presentes na linguagem (aritmética, chamada de função, etc).
- **array:** regra que define o formato do array na linguagem.
- **array_literal:** regra que define um array literal. Ex.: {1, 2, 3}.
- **function_call:** regra que define o formato da chamada de uma função.
- **type:** regra que define os tipos presentes na linguagem.

No mais, a linguagem deverá suportar os tipos 'int', 'float' e 'string'. Também deverá suportar comentários de uma linha (//) e de bloco (/* */).

DICA: olhem e analisem bem as imagens das saídas esperadas para entender a estrutura e o formato da linguagem que estamos pedindo.