

Compte-Rendu

Projet compilation

Novembre 2025



KERAVEL Nino

COUTURE Jules

Sommaire

1. Ce qui a été implémenté	2
2. Non implémenté	2
3. Implémenté mais ne fonctionne pas	3

1. Ce qui a été implémenté

A chaque fois il y a “fonctionnalité (*test correspondant*)”

- Structure de base lexer -> parser -> optimizer (vide) -> codegen
- Déclaration de variables : juste int
- Variables masquées (*test_contexts*)
- Boucles (*test_base_loop.c*)
- Boucles imbriquées (*test_nested_different_loops*, *test_nested_for_loops*)
- Condition (ici imbriquée dans une loop) (*test_conditions*)
- Fonctions et récursion (*test_fibo_rec* *test_fibo_dyn*)
- Pointeurs, malloc, tableaux (*test_base_tabs*)
- Inclusion d'une mini std (*malloc print println*)

2. Non implémenté

- Variable avec _
- Déclaration+initialisation sur la même ligne (int i = 0)
- Opérateurs d'incrémantation et de décrémantation

Les deux derniers n'ont pas implémenté car c'est surtout du sucre syntaxique mais c'est vrai que ça aurait pu être utile, on le voit sur les tests ça donne une écriture vraiment verbeuse même pour une simple boucle for

- Fonction input() pour lire l'entrée utilisateur
- Commentaires

C'est dû à une mauvaise détection des changements de ligne, c'est pour cela que même si on a voulu implémenter des erreurs qui indique la ligne de l'erreur, une fois sur

deux cela ne marche pas donc on a préféré ne pas tester les commentaires non plus.

3. Implémenté mais ne fonctionne pas

- Système d'erreurs

Problème décrit précédemment avec les numérotation de ligne (et colonnes) + ajout de la std en début du code (et pas en utilisant un linker à la fin) fait que de toute façon la ligne sera mauvaise.