

Compte-Rendu

Projet compilation

14/11/2025



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)

1. Non implémenté

- variable avec _
- declaration+initialisation sur la même ligne (int i = 0)
- opérateurs d'incrément et de décrémentation

Les deux derniers n'ont pas été implémentés 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 2
ça ne marche pas donc on a préféré ne tester les
commentaires non plus

2. 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.