# Rapport du projet de Compilateur

# 1 But

Ce projet du cours de compilation du module de 3ème année de développement système a pour but de développer un interpréteur de c simplifié.

# 2 Spécification

Mon interpréteur reconnait les types simples « int » et « float ».

## 2.1 Dépendances

Langage source : **C simplifier**

Langage intermédiaire : **Python 3.5.0** avec

• pydot 1.0.3

• PLY 3.8 (avec les modules lex et yacc)

## 2.2 Type acceptés

Int

Float

## 2.3 Déclarations acceptées

Int b ;

Int b = 0 ;

## 2.4 Vérification des types

L’interpréteur génère un warning si :

* Une variable est déclarer mais pas utilisée ;
* Une variable est utilisée sans avoir été déclarée.

L’interpréteur génère une erreur sur la ligne courante si :

* Une variable est utilisée sans avoir été initialisée.

# 3 Méthode

# 5 Prise en main

Notre compilateur est composé de quatre modules distincts :

* Analyseur lexical (lex.py) ;
* Analyseur syntaxique (parser.py) ;
* Constructeur d’arbre syntaxique (AST.py) ;
* Analyseur sémantique (semantic.py) ;
* Interpréteur récursif (recInterpreter.py)

# 6 Conclusion

Ce projet a été intéressant. Il nous a permis de mieux nous apercevoir de la réelle complexité de faire un « vrai » compilateur pour un vrai langage de programmation. Il manque plus de la moitié des règles de grammaire à traiter pour le langage C, mais ce projet a déjà eu une complexité non négligeable. Le plus gros souci a été, sans doute, de suivre au maximum la grammaire officielle du langage tout en supprimant les instructions qui n’étaient pas à traiter et en les adaptant à nos nécessités.

Il reste à souligner que les travaux pratiques réalisés antérieurement pendant les cours nous ont donné un bon guide de quoi et comment faire les choses. Sans ceux-ci, je ne pense pas que ce projet serait arrivé jusqu’où nous sommes arrivés.