

## RAPPORT INTERPRÉTATION ET COMPILATION

### Compilateur Mini Python vers MIPS en Racket



```
mov    rax, 0x0a687f
push   rax
xor     rax, rax
mov     rsi, rsp
mov     rdi, 1
mov     rdx, 8
call   write
```

*Suivi par Pablo RAUZY*

- **Nom** : TACHOUR
- **Prénom** : Omar
- **Num d'étudiant** : 16706265

## Présentation du projet :

Mon projet consiste à coder un compilateur d'un sous-ensemble du langage Python vers l'assembleur MIPS en utilisant un langage fonctionnel (Racket) et simulateur SPIM.

## Les instructions python possibles à interpréter par mon projet :

- Les opérations arithmétiques : (**addition, soustraction, multiplication, division et modulo**) .
- Les opérations logiques: (**and or xor not**) .
- Les opérations de comparaison : ( **> < ≥ ≤ == !=** ) .
- Déclarations de variables.
- Les mots clés prédéfinis : **False** et **True**.
- Les conditions (**if**).
- Les boucles (**while**).
- Affichage des chaînes des caractères, chiffres, variables .
- Déclaration de fonctions.
- Appel aux fonctions.
- Insertion des commentaires

## Les fichiers constituant mon projet :

- « ast.rkt » : contient les structures de l'arbre de syntaxe abstraite
- « parser.rkt » : comporte les outils nécessaires à l'analyse syntaxique.
- « compiler.rkt » : consiste à traduire les instructions du langage source en instructions MIPS.
- « lexer.rkt » : contient les outils nécessaires pour l'analyse lexicale qui transforme le code en lexèmes.
- « mips.rkt » : il contient l'assembleur MIPS.