



**École Supérieure Polytechnique**  
**Département Génie Informatique**  
**Diplôme d'Ingénieur de Conception (DIC), 2<sup>e</sup> année**  
**Année Universitaire 2022-2023**  
**Mini projet de Compilation : Analyseur Lexical Lex**

*Dr Mandicou BA*

## **Tâche 1 :**

L'objectif de ce mini projet est de programmer un analyseur lexical pour le langage M. L'analyseur lexical se présente comme une fonction, `int yylex()` qui, à chaque appel, renvoie comme résultat le code de l'unité lexicale dont c'est le tour. De plus, cette fonction met dans la variable `yytext` la valeur de l'unité lexicale reconnue. Ainsi, lorsque `yylex` reconnaît la chaîne de caractères `123`, elle renverra le type `NOMBRE` et mettra dans la variable `yytext` la chaîne de caractères `«123»`.

## **1 Description du Langage M**

Le langage M est un langage de programmation minimaliste, inspiré du langage C.

**Types** : Le langage M connaît deux types de variables :

- Un type simple : le type entier.
- Un type dérivé : les tableaux d'entiers, déclarés avec des crochets [...]

**Opérateurs** : Le langage L connaît les opérateurs suivants :

- arithmétiques : `+`, `-`, `*`, `/`
- comparaison : `<`, `=`
- logiques : `&` (et), `|` (ou), `!` (non)

**Instructions** : Le langage M connaît les instructions suivantes :

- Instruction vide ;
- Bloc d'instructions, délimité par des accolades
- Affectation `a = b + 1 ;`
- Instruction `si expression alors ... et si expression alors ... sinon ...`
- Instruction `tantque expression faire ...`
- Instruction `retour expression ;`
- Instruction d'appel à fonction `fonction( liste d'expressions ) ;`

**Fonctions** : un programme M est une suite de fonctions, dont *main*

- Ce sont des fonctions à résultat entier.
- Le passage des arguments se fait par valeur.
- Les fonctions possèdent des variables locales.
- Une fonction ne peut pas être déclarée à l'intérieur d'une autre.
- On peut ignorer le résultat rendu par une fonction.

### Fonctions prédéfinies :

Les entrées-sorties de valeurs entières se font à l'aide de deux fonctions prédéfinies,  $a = lire()$ ; et  $ecrire(a)$ ;

### Commentaires et blancs

Les unités lexicales peuvent être séparées les unes des autres dans le programme source par un nombre quelconque de blancs, caractères de tabulation et retours chariot . Tous ces caractères doivent être ignorés. C'est l'analyseur lexical qui doit s'occuper de supprimer les espaces. Cependant, l'analyseur doit s'en servir pour séparer les unités lexicales

### Les différentes type d'unité lexicales considérées :

On distingue 5 types d'unités lexicales :

- les symboles simples (+, ;, (, {, ...),
- les mots-clefs, dont :
  - les instructions de contrôle (si, alors, retour, ... ),
  - les types (entier),
  - les fonctions spéciales *lire* et *écrire*,
- les noms de variables,
- les noms de fonctions et
- les nombres, uniquement entiers.

## 2 Travail demandé

1. Il est demandé de réaliser un tel analyseur lexical avec l'outil *lex*.
2. Il faut rendre un rapport rédigé en LaTeX, expliquant clairement votre projet ainsi que toutes les illustrations nécessaires. Rendre aussi les codes sources clairement écrits, avec les commentaires nécessaires.
3. Possibilités de travailler en binôme
4. A rendre avant le 14/07/2023, à 23h59min sur la plateforme <https://m1glslsi.mandicouba.net>, au niveau du répertoire *ProjetCompilation*. Il faut impérativement déposer le rapport et les codes sources.