Soit le programme suivant écrit dans un langage, auquel nous chercherons à créer un analyseur syntaxique.

|  |  |
| --- | --- |
| debut  Var = 1; c = Var; a=4 ;  fin | **Code 1** |
|  |

# A. Analyse syntaxique

1. Définir les symboles terminaux du langage et les symboles non-terminaux.

les symboles terminaux : **var , c , a ,4, 1**

les symboles non-terminaux : **debut , = , ; , fin**

1. Définir un élément de départ

**debut**

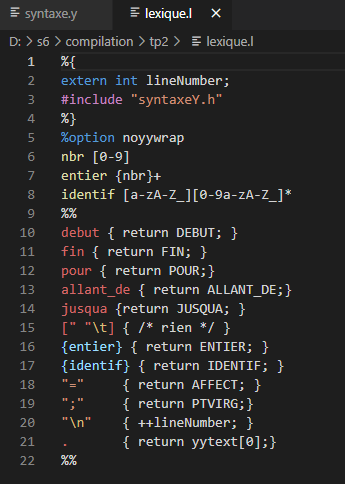
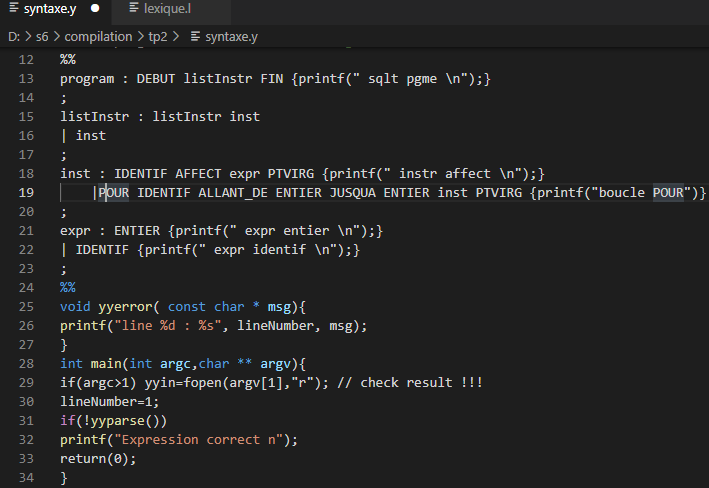
1. Définir les règles de réécriture

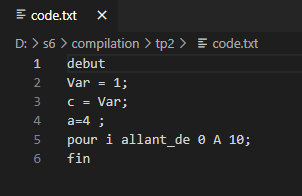
**debut**

**//des instructions**

**fin**

# Création d’analyseur syntaxique

* Fichiers syntaxe.y et lexique.l
* Le fichier text

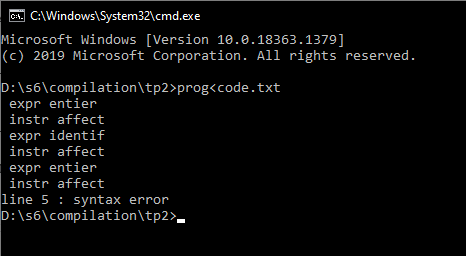


* Les fichiers générés par les deux commandes

**flex -olexiqueL.c lexique.l :** génère le fichier **lexiqueL.c**

**bison -d -osyntaxeY.c syntaxe.y :** génère le fichier **syntaxeY.c**

* Test de l’analyseur syntaxique.



**Conclusion :**

L’analyse lexicale nous permit de définir notre propre grammaire à respecter par l’utilisateur, et aussi les unités lexicales sont génère automatiquement par bison.