|  |  |
| --- | --- |
| Début {  ma\_Varaibale = 12;  b = 2; si(var1!=b)alors{  var1 = b;  }  fin | **Code 1** |

# A. L’analyse lexical

1. Quels sont les mots clé de ce langage ?

**Debut**

**ma\_variable , b , var1**

**=**

**12 , 2**

**;**

**si**

**(**

**!=**

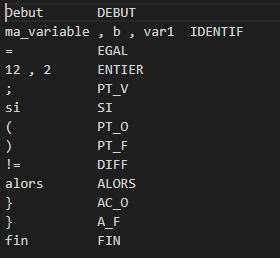
**)**

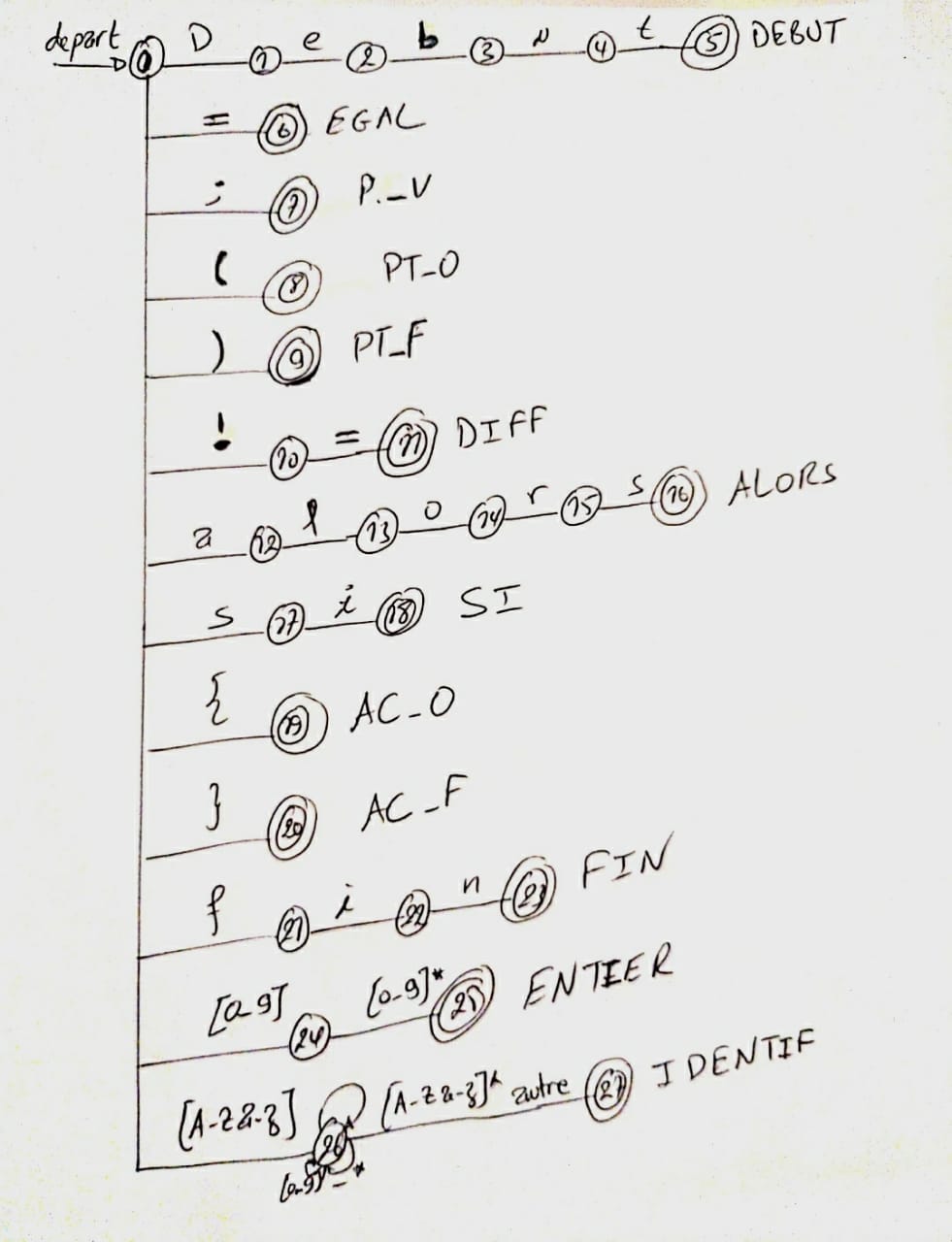
**alors**

**{**

**}**

**fin**

1. Définir les tokens correspondants aux mots clé de la question 1. 
2. Donner les automates reconnaissants les tokens dans ce langage



1. Donner les expressions régulières qui permettent de définir les tokens de ce langage

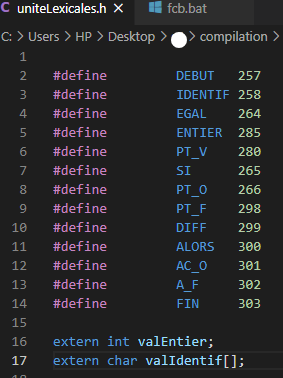
**nbr [0-9]**

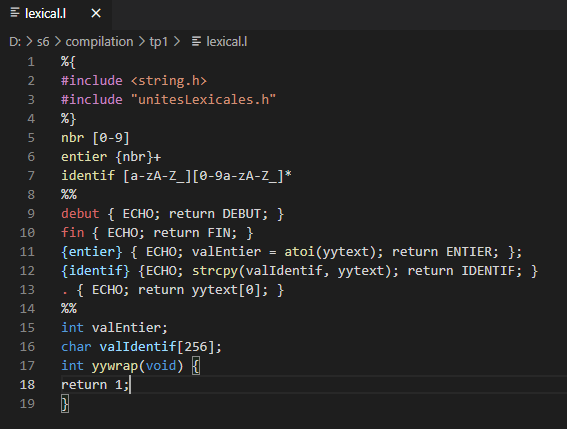
**entier {nbr}+**

**identif [a-zA-Z\_][a-zA-Z\_0-9]\***

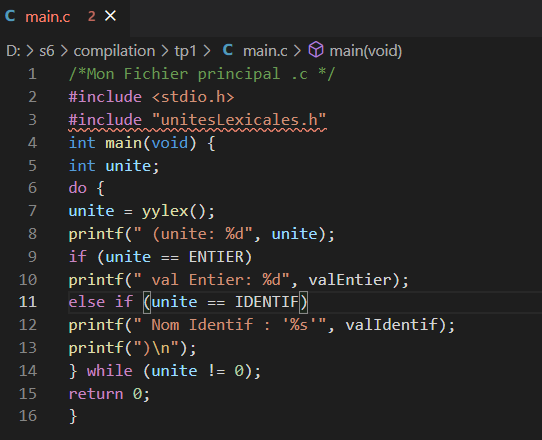
# Création d’un analyseur lexical

1-2 – Fichier UNITESLEXICALES.h :

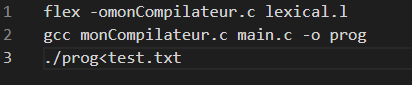


1. Créer un fichier .l et recopier le **Code 2.** 4- Créer un fichier .c et recopie le **Code 3.**

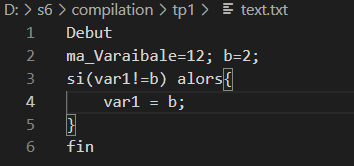
4- Créer un fichier .c et recopie le **Code 3.**



5- Lancer l’invite de commande et taper les commandes suivantes pour générer l’analyseur lexical



1. Recopier le code du programme **Code 1** dans fichier .txt et taper la commande suivante pour tester votre analyseur



Après l’exécution de toutes les commandes, on a trouvé le résultat suivant :

