

Exercice 1 :

On considère la syntaxe suivante :

Syntaxe	Exemple1	Exemple2	Exemple3	Exemple4
Id = valeur	Code	Code	Code	Code
Id = Id + Id	a=10	a = b + c	a=b+20	a=b*c+d+e
Id = Id + valeur				

Partie I :

On propose la grammaire suivante:

S = id = T E
 T = id | valeur
 E = + T E | *T E | ε

1. Donner l'ensemble V_N et l'ensemble V_T .
2. Donner les premiers et les suivants de chaque symbole non terminal.
3. Élaborer la table d'analyse LL de cette grammaire.
4. Analyser la phrase $x = y + 4 + z$ et Donnez l'arbre de dérivation associé. A noter que x,y et z sont des identificateurs.

Partie II :

Cette partie traitera une grammaire permettant de produire la syntaxe d'une fonction sous Python.

On considère la syntaxe suivante:

Syntaxe	Exemple1	Exemple2	Exemple3
def fonction(liste_paramètres): instr où instr présente l'instruction de la partie I	Code def fct1(x): x=x+x*x	Code def fct2(x,y): x=x+2*y	Code def fct1(x,y,z): x=x+2*y+z
	Résultat de l'exécution	Résultat de l'exécution	Résultat de l'exécution
	>>> x=1 >>> fct1(x) >>>x 2	>>> x=1 >>> y=2 >>> fct2(x,y) >>>x 5 >>>y 2	>>> x=1 >>> y=2 >>> z=3 >>> fct3(x,y,z) >>>x 8 >>>y 2 >>>z 3

1. Proposer une grammaire pour la syntaxe des fonctions.
Indication : pensez à utiliser la partie I;
2. Donner l'ensemble V_N et l'ensemble V_T de la nouvelle grammaire
3. Donner les premiers et les suivants de chaque symbole non terminal.
4. Élaborer la table d'analyse LL de cette grammaire.

Exercice 2

Un dictionnaire en Python est une structure des données qui utilisent un système d'indexation (clé) propre à lui, chaque clé est séparée de sa valeur par deux points (:), les éléments sont séparés par des virgules, et le tout est enfermé dans des accolades.

Un dictionnaire vide sans aucun article est écrit avec seulement deux accolades, comme ceci: {}.

	Script	résultat
Exemple1	>>> dict={'a':1,4:"bonjour",3.14:"pi",3:22/7}	>>> dict['a'] 1
Exemple2	>>> dict={'a':1,4:"bonjour",3.14:"pi",3:22/7}	>>> dict[4] 'bonjour'
Exemple3	>>> dict={'a':1,4:"bonjour",3.14:"pi",3:22/7}	>>> dict[3.14] 'pi'
Exemple4	>>> dict={'a':1,4:"bonjour",3.14:"pi",3:22/7}	>>> dict[3] 3.142857142857143

Partie I Analyse ascendante:

On propose la grammaire suivante:

$S \rightarrow id = \{ C \}$

$C \rightarrow id: val \mid id: val , C$

où id désigne un identificateur, et val désigne une valeur de type quelconque

1. Donner l'ensemble V_N et l'ensemble V_T .

$V_N = \{ \dots \} ; V_T = \{ \dots \} ;$

2. Donner les premiers et les suivants de chaque symbole non terminal.

Premier= { }	Suivant= { }
--------------	--------------

3. Élaborer la table d'analyse LL de cette grammaire

.....
.....
.....
.....
.....

4. Analyser l'instruction $d = \{ 'a':1 , 4: 3 \}$ et donnez l'arbre de dérivation correspondant.

Analyse			Arbre
Pile	Entrée	Sortie	

Partie II :
 La deuxième partie sera dédiée à une grammaire permettant de produire un dictionnaire contenant d'autres dictionnaires

Exemple1	Exemple2
<pre>>>> d1 = {'a':1 , 4: 3.14, 7 : 9} >>> d2 = {1 : {'b':2 , 6 : "smi"} , 4: 3.14, 7 : 9} >>> d2[1] {'b': 2, 6: 'smi'} >>> d2[1]['b'] 2 >>> d2[7] 9</pre>	<pre>>>> d1 = {'a':1 , 4: 3.14, 7 : 9} >>> d2 = {1 : {'b':2 , 6 : "smi"} , 4: 3.14, 7 : 9} >>> d3 = {2 : d2 , 8: d1, 7 : 9} >>> d3[2] {1: {'b': 2, 6: 'smi'}, 4: 3.14, 7: 9} >>> d3[2][1] {'b': 2, 6: 'smi'} >>> d3[2][1][6] 'smi'</pre>

1. Proposer une grammaire pour la syntaxe permettant de produire un dictionnaire de dictionnaires

.....

.....

.....

.....

.....

.....

.....

2. Trouver l'ensemble V_N et l'ensemble V_T de la nouvelle grammaire

$V_N = \{ \}$	$V_T = \{ \}$
---------------	---------------

3. Trouver les premiers et les suivants de chaque symbole non terminal.

Premier={ }

Suivant={ }

4. Élaborer la table d'analyse LL de cette grammaire?

.....
.....
.....
.....
.....
.....
.....