

Test Informatique N°1

Exercice 1: Mot de Dyck

Dans cet exercice, on se propose avoir créer le module *pile.py* qui implémente la structure Pile où on a défini les fonctions suivantes :

- *creer_pile()* permettant de créer et retourner une pile vide.
- *empiler(p,e)* permettant d'ajouter un élément *e* à la pile *p*.
- *depiler(p)* permettant de dépiler et retourner le sommet de la pile *p*.
- *est_vide(p)* à résultat booléen, permettant de vérifier si la pile *p* est vide.

On appelle mot de **Dyck** tout mot construit à partir de deux lettres **lettre1** et **lettre2** vérifiant les deux propriétés suivantes :

1. Le mot doit commencer par **lettre1** et contenir autant de la **lettre1** que de la **lettre2**.
2. Dans tout préfixe du mot, le nombre d'occurrences de **lettre1** est supérieur ou égal au nombre d'occurrences de **lettre2**, c'est-à-dire lorsqu'on fait le parcours du mot du gauche à droite on a toujours, le **nombre_lettre1** $\geq$ **nombre_lettre2**

Exemple : Pour lettre1='E' et lettre 2='D' :

- Le mot 'EDED' est un mot Dyck car dans tous ses préfixes ; qui sont 'E', 'ED', 'EDE' et 'EDED' ; on a *nombre_lettre1* $\geq$ *nombre_lettre2*.
- Le mot 'EEDD' n'est pas un mot Dyck car dans le préfixe 'EED' on a *nombre_lettre1* $>$ *nombre_lettre2*.
- Le mot 'EDDE' n'est pas un mot Dyck car dans le préfixe 'EDD' on a *nombre_lettre2* $>$ *nombre_lettre1*.

Travail demandé :

On veut tester si un mot donné est un mot de Dyck par différentes méthodes. Pour cela on définira plusieurs fonctions dont chacune prend trois paramètres : une chaîne de caractère **mot**, supposé de taille paire et composé uniquement des lettres **lettre1** et **lettre2**, ainsi que deux caractères **lettre1** et **lettre2** et qui retourne True si mot est un mot de Dyck et False sinon.

1. Écrire une fonction *est_Dyck1(mot, lettre1 ,lettre2)* qui vérifie si mot est un mot de Dyck.
2. Écrire une fonction récursive *DyckR(mot, lettre1 ,lettre2)* qui vérifie si mot est un mot de Dyck.
3. Écrire une fonction *est_Dyck2(mot, lettre1 ,lettre2)* qui vérifie si mot est un mot de Dyck en utilisant une pile.

NB : Penser au principe de résolution du problème permettant de vérifier si une chaîne est bien parenthésée.