

TP noté 2 – 50 minutes

Université Paris-Diderot

Pour ce second TP noté, aucun squelette n'est fourni, vous devrez donc créer votre fichier de code Python vous-même, puis me l'envoyer par mail lorsque vous aurez fini à l'adresse `cyprien.mangin@m4x.org` ou `mangin.cyprien@gmail.com`.

Vous avez un ordinateur avec Python à votre disposition, *servez-vous en* pour avoir un fichier qui s'exécute sans erreur à la fin !

Exercice 1 (Variable globale)

1. Définir une variable globale nommée `texte` et qui contient au départ une chaîne de caractères vide `""`.
2. Définir une procédure `afficherTexteActuel` qui ne prend aucun argument et affiche le contenu de la variable `texte`.
3. Définir une procédure `concatener` qui prend une chaîne de caractères en argument, et la concatène à la variable `texte`. Par exemple, si `texte` vaut `"Hello "`, alors après l'appel `concatener("world")`, la variable `texte` doit valoir `"Hello world"`.
4. Définir une procédure `vider` qui ne prend aucun argument, et réinitialise la variable `texte` pour qu'elle contienne une chaîne de caractères vide.

□

Exercice 2 (Diagonales)

Écrire une fonction `diagonales` qui prend en paramètre une liste de listes `t` telle que pour tout `i`, `t[i]` vérifie `len(t[i]) == len(t)` (il s'agit donc d'une matrice carrée), et renvoie ses deux diagonales dans une liste de listes.

Contrat:

Par exemple :

— `diagonales([[0, 1, 2], [3, 4, 5], [6, 7, 8]])` renvoie `[[0, 4, 8], [6, 4, 2]]`.

— `diagonales([[1, 0, 0, 2], [0, 1, 2, 0], [0, 2, 1, 0], [2, 0, 0, 1]])` renvoie `[[1, 1, 1, 1], [2, 2, 2, 2]]`.

Visuellement, les deux matrices utilisées en exemple sont :

0	1	2		1	0	0	2
3	4	5		0	1	2	0
6	7	8	et	0	2	1	0
				2	0	0	1

□

Exercice 3 (Quantificateurs)

1. Chacune des 3 fonctions ci-dessous prend une liste de listes d'entiers t en argument et renvoie un booléen. Indiquer pour chacune d'entre elles quelle spécification parmi les 4 suivantes lui correspond :
 - (A) La fonction renvoie True si et seulement si t contient au moins une liste qui contient au moins un 0.
 - (B) La fonction renvoie True si et seulement si t contient au moins une liste qui ne contient que des 0.
 - (C) La fonction renvoie True si et seulement si t ne contient que des listes qui contiennent au moins un 0.
 - (D) La fonction renvoie True si et seulement si t ne contient que des listes qui ne contiennent que des 0.(Pour répondre, ajoutez dans votre code un commentaire comme : `# fonction1 vérifie (A).`)

```
1 def fonction1(t):
2     for i in range(0, len(t), 1):
3         for j in range(0, len(t[i]), 1):
4             if t[i][j] != 0:
5                 return False
6     return True
7
8 def fonction2(t):
9     for i in range(0, len(t), 1):
10        b = False
11        for j in range(0, len(t[i]), 1):
12            if t[i][j] == 0:
13                b = True
14        if not b:
15            return False
16    return True
17
18 def fonction3(t):
19     for i in range(0, len(t), 1):
20         for j in range(0, len(t[i]), 1):
21             if t[i][j] == 0:
22                 return True
23    return False
```

2. Écrire fonction4, qui prend une liste de listes d'entiers en argument, renvoie un booléen, et vérifie la spécification qui n'était vérifiée par aucune des fonctions ci-dessus.
3. Écrire la fonction plusDeQuantificateurs, qui prend une liste de listes de chaînes de caractères t en argument et renvoie un booléen qui vaut True si et seulement si t contient au moins une liste de chaînes de caractères $t[i]$ telle que tous les éléments de $t[i]$ contiennent au moins une fois le caractère "e".

Contrat:

Par exemple :

- plusDeQuantificateurs([["."], ["Hi", "there"]]) renvoie False.
- plusDeQuantificateurs([["."], ["Hello", "there"]]) renvoie True parce que toutes les chaînes de caractères de la liste $t[1]$ contiennent au moins un "e".
- plusDeQuantificateurs([["."], []]) renvoie True.

□