

EXERCICES SUITE AU TP SUR LES SÉQUENCES 1

COMPREHENSION DE LISTE

Q1- Quelle instruction permet d'affecter la liste `[0,1,4,9,16]` à la variable `tableau` ?

Réponses :

A- `tableau = [i**2 for i in range(4)]`

B- `tableau = [i**2 for i in range(5)]`

C- `tableau = [i**2 for i in range(16)]`

D- `tableau = [i**2 for i in range(17)]`

Q2- Que vaut l'expression `[2 * k for k in range(5)]` ?

Réponses :

A- <code>[0,2,4,6,8]</code>	C- <code>[1,2,4,8,16]</code>
B- <code>[2,4,6,8,10]</code>	D- <code>[2,4,8,16,32]</code>

TABLEAUX 2 DIMENSIONS

Q3- On définit : `matrice = [[1,2,3], [4,5,6], [7,8,9], [10,11,12]]`

Quelle est la valeur de `matrice[1][2]` ?

Réponses :

A- 2	C- 6
B- 4	D- 8

Q4- n définit la liste L ainsi : `L = [[1], [1,2], [1,2,3]]`

Des égalités suivantes, une seule est fausse. Laquelle ?

Réponses :

A- <code>len(L[0]) == 1</code>	C- <code>len(L[2]) == 3</code>
B- <code>len(L) == 6</code>	D- <code>L[2][2] == 3</code>

Q4- On considère la liste de listes suivante :

```
tictactoe = [['X', 'O', 'O'],
             ['O', 'O', 'O'],
             ['O', 'O', 'X']]
```

Quelle instruction permet d'obtenir une diagonale de 'X' ?

Réponses :

A- <code>tictactoe[3] = 'X'</code>	C- <code>tictactoe[1][1] = 'X'</code>
B- <code>tictactoe[4] = 'X'</code>	D- <code>tictactoe[2][2] = 'X'</code>

Q5- On définit ainsi une liste M :

```
M = [['A', 'B', 'C', 'D'], ['E', 'F', 'G', 'H'], ['I', 'J', 'K', 'L']]
```

Quelle expression vaut la chaîne de caractères 'H' ?

Réponses :

A- M[1][3]

B- M[3][1]

C- M(7)

D- M(8)

Q6- On représente un plateau de jeu d'échec par une liste de listes dans laquelle on place des 1 pour représenter une case où se trouve une tour et des 0 pour représenter les cases vides.

Par exemple le code rci-dessous eprésente la situation de la figure ci-contre.

```
echiquier[2][0] = 1
echiquier[3][1] = 1
```

	0	1	2	3	4	5	6	7
0								
1								
2	♖							
3		♜						
4								
5								
6								
7								

Deux tours sont en prise si elles se trouvent sur une même ligne ou sur une même colonne.

Parmi les codes suivants, lequel permet de vérifier que la tour placée en ligne i et en colonne j n'est en prise avec aucune tour placée dans les colonnes à sa gauche ?

Réponses :

A- <pre>def ok(echiquier,i,j): for col in range(i): if echiquier[i][col] == 1: return False return True</pre>	C- <pre>def ok(echiquier,i,j): for col in range(j): if echiquier[i][col] == 1: return False return True</pre>
B- <pre>def ok(echiquier,i,j): for lig in range(i): if echiquier[lig][j] == 1: return False return True</pre>	D- <pre>def ok(echiquier,i,j): for lig in range(j): if echiquier[lig][j] == 1: return False return True</pre>

ALGORITHME D'INVERSION DE LISTE

Q7. Compléter le script de la fonction inverse

```
def inverse(L1):
    L2 = []
    for i in range(len(L1)):
```

```

        L2.append(L1[-i-1])
    return L2
> inverse(['C','O','U','R','I','R'])
['R', 'I', 'R', 'U', 'O', 'C']

```

Q8. Ecrire une fonction `palindrome` qui prend en paramètre une chaîne de caractères et qui retourne `True` si c'est un palindrome, c'est-à-dire s'il peut se lire dans les 2 sens

```

def palindrome(mot):
    L1 = list(mot)
    ...
    ...
    ...

> palindrome('lustucru')
False
palindrome('ressasser')
True

```

ALGORITHME DE TRI PAR INSERTION

Q9. Compléter le script

```

def tri1(table):
    for k in range(1,len(table)):
        temp = table[k]
        j = k
        while j>0 and ...
            table[ .. ]=table[ .. ]
            j-=1
        table[j]=temp
    return table

```

Q10. Compléter la liste à chaque étape du tri

etape	liste à la fin de l'étape
0	[49, 4, 89, 13, 44, 60, 12, 2, 5, 62]
1	...
2	
3	
4	
5	
6	
7	
8	
9	