

Rappels avant d'aborder les algorithmes de tri

tableaux, listes et tuples

- Tableau : `mon_tableau = array([12, 15, 11, 18])` // ne peut pas être modifié en taille une fois créé, mais on peut modifier les valeurs des cases. Rarement utilisé en Python
- tuples : `mon_tuple = (2,6,5,8,9)` //ne peut pas être modifié (non mutable)
- Listes : `ma_liste = [2,6,8,9,7,5,8]` // peut être modifié en valeur et en taille (une liste est mutable)

Création de la liste :

Créer une liste vide : `MaListe = []` **Créer une liste contenant des valeurs** : `MaListe = [1,2,3,6,9]`
Créer une liste par compréhension : `MaListe = [p for p in range(0, 5)]` donne `[0,1,2,3,4]`

Afficher une valeur de la liste :

Les éléments composant la liste sont « repérés » par l'index du tableau. Ainsi : `Print(MaListe[0])` affiche **1**

Modification de la liste :

Pour **ajouter** une case en queue de liste : `append`

`MaListe.append(8)` transformera notre liste en `[1,2,3,6,9,8]` (on a ajouté la valeur 8)

Pour **modifier** une valeur dans la liste :

`MaListe[0]=15` transformera notre liste en `[15,2,3,6,9,8]`

Compter le nombre d'items d'une liste : `len(MaListe)` // renvoi 6 pour notre exemple

Pour supprimer une case n'importe où dans la liste : `del`

`del MaListe[3]` transformera notre liste en `[15,2,3,9,8]`

Pour **copier une liste** : on peut utiliser la fonction `deepcopy` du module `copy` :

`import copy` // à placer en tout début du programme

`x = [[1,2,3,2,6]]` // création de la liste x

`y = deepcopy(x)` // y sera une copie par valeur de x

`z = x` // z sera une copie par référence de x : toute modif sur z entrainera aussi une modif sur x

Exercices d'algorithmique manipulant une liste avec une boucle :

Ecrire un algorithme qui permet de trouver la plus grande valeur d'une liste (on prendra `t=[2,6,9,78,4]`, et `k` pour nom de variable contenant à la fin la plus grande valeur). Implanter en Python pour vérifier votre algorithme.

Ecrire un algorithme qui permet de trouver à quel indice se trouve la plus petite valeur d'une liste (on prendra `t=[2,6,9,78,1,4]`, et `min` pour nom de variable contenant à la fin l'indice de la case contenant la plus petite valeur). Implanter en Python pour vérifier votre algorithme.

Ecrire un algorithme qui permet de trouver la plus petite valeur d'une liste et de supprimer la case contenant cette valeur. Implanter en Python pour vérifier votre algorithme.



Questions sup (QCM):

1/ On considère le script suivant :

```
t = [2, 8, 9, 2]
t[2] = t[2] + 5
```

Quelle est la valeur de t à la fin de son exécution ?

Réponses :

A- [2, 13, 9, 2] B- [2, 8, 14, 2] C- [7, 13, 14, 7] D- [7, 13, 9, 2]

2/On dispose dans le tableau annee2019 les températures mensuelles moyennes d'une région française. On exécute le script suivant :

```
annee2019 = [('janvier',6), ('février',6), ('mars',12), ('avril',20), ('mai',23), ('juin',25), ('juillet',29),
('août',25), ('septembre',22), ('octobre',15), ('novembre',11), ('décembre',7)]
m = annee2019[0][1]
for mois in annee2019:
    if (m > mois[1]):
        m = mois[1]
```

Que contient la variable m à la fin de cette exécution ?

Réponses :

A- le mois le plus froid B- le mois le plus chaud C- la température moyenne la plus basse
D- la température moyenne la plus haute

3/ - 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- tictactoe[3] = 'X' B- tictactoe[4] = 'X' C- tictactoe[1][1] = 'X' D- tictactoe[2][2] = 'X'

4/ - On exécute le code suivant :

```
A = [ [1,2,3], [4,5,6], [7,8,9] ]  
B = [ [0,0,0], [0,0,0], [0,0,0] ]  
for i in range(3):  
    for j in range(3):  
        B[i][j] = A[j][i]
```

Que vaut B à la fin de l'exécution ?

Réponses :

A- rien du tout, le programme déclenche une erreur d'exécution

B- [[3,2,1], [6,5,4], [9,8,7]] C- [[1,4,7], [2,5,8], [3,6,9]] D- [[7,8,9], [4,5,6], [1,2,3]]

5/ - On considère la fonction suivante :

```
def somme(tab):  
    s = 0  
    for i in range(len(tab)):  
        .....  
    return s
```

Par quelle instruction faut-il remplacer les points de suspension pour que l'appel `somme([10,11,12,13,14])` renvoie 60 ?

Réponses :

A- `s = tab[i]` B- `s = s + tab[i]` C- `tab[i] = tab[i] + s` D- `s = s + i`

6/ - On considère deux entiers strictement positifs L et C. On note $n = L \cdot C$ leur produit et on écrit la fonction suivante, qui construit un tableau de L lignes et C colonnes, contenant les entiers consécutifs de 0 à n-1 :

```
def construitTable(L,C):  
    t = []  
    for i in range(L):  
        ligne = []  
        for j in range(C):  
            .....  
        t.append(ligne)  
    return t
```

Par exemple, l'appel `construitTable(2,3)` doit renvoyer la table : `[[0, 1, 2],[3, 4, 5]]`

Que faut-il écrire à la place des points de suspension pour obtenir ce résultat ?

Réponses :

A- `ligne.append(i + C*j)` B- `ligne.append(L*i + j)` C- `ligne.append(i + L*j)` D- `ligne.append(C*i + j)`

7/ - Quelle est la valeur de l'expression `[[i for i in range(5)] for j in range(3)]` ?

Réponses :

A- `[[0, 1, 2], [0, 1, 2], [0, 1, 2], [0, 1, 2], [0, 1, 2]]`

B- `[[0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4]]`

C- `[[0, 1, 2, 3], [0, 1, 2, 3], [0, 1, 2, 3], [0, 1, 2, 3], [0, 1, 2, 3]]`

D- `[[0, 1, 2, 3, 4, 5], [0, 1, 2, 3, 4, 5], [0, 1, 2, 3, 4, 5]]`

8/ Quel est le contenu du tableau référencé par la variable `mon_tab` après l'exécution du programme ci-dessous ?
(implanter en python pour vérifier votre réponse)

```
l = [1, 7, 9, 15, 5, 20, 10, 8]
```

```
mon_tab = [p**2 for p in l if p < 10]
```

Rappel : `p**2` permet d'obtenir la valeur de p élevée au carrée