

Exercices d'application

Exercice 1 : Indigage et découpe d'une liste

On exécute l'instruction suivante sur la console python

```
>>>a=[0,2,4,8,3,5,6,7]
```

x **Compléter** le tableau suivant en indiquant pour chaque expression les valeurs renvoyées par la console et vis versa

Expression	a[0]	a[1]	a[-1]	a[-2]
Valeur renvoyée	0	2	7	6

Expression	a[:2]	a[3:6]	a[-3:]	a[2:4]
Valeur renvoyée	[0,2]	[8,3,5]	[5,6,7]	[4,8]

Exercice 2 : Analyse d'une liste

Le programme suivant est exécuté:

```
1 a=[1,7,5,3]
2 for i in range(1,len(a)):
3     if a[i]<a[i-1]:
4         a[i],a[i-1]=a[i-1],a[i]
5 resultat=a[-1]
```

x **Indiquer** la valeur renvoyée par l'instruction len(a).

len(a) renvoie 4

x **En déduire** le nombre de fois que sera exécuté le bloc d'instruction associé à la boucle for.

La boucle for sera exécutée 4 fois

x **Compléter** la table de séquence suivante en indiquant pour chaque étape du programme :

- le numéro de l'instruction exécutée
- l'état de liste référencée par a

- les valeur de a[i] et a[i-1]
- le résultat de l'expression a[i]<a[i-1]

Etape	Ligne exécutée	Valeur de i	Valeur de a[i]	Valeur de a[i-1]	Résultat de a[i]<a[i-1]	Etat de a
1	1	X	X	X	X	[1,7,5,3]
2	2	1	7	1	f	[1,7,5,3]
3	3	1	7	1	f	[1,7,5,3]
4	2	2	5	7	t	[1,7,5,3]
5	3	2	5	7	t	[1,7,5,3]
6	4	2	7	5	f	[1,5,7,3]
7	2	3	3	7	t	[1,5,7,3]
8	3	3	3	7	t	[1,5,7,3]
9	4	3	7	3	f	[1,5,3,7]

x **Déterminer** la valeur de la variable resultat à la fin de l'exécution du programme.

Resultat contiendra 7

x Ce programme recherche la valeur la plus grande d'une liste. **Modifier** ce programme afin qu'il recherche la valeur la plus petite.

Il faut remplacer < par >

Exercice 3 : Traitement de données

Le programme suivant traite les données contenues dans la liste a :

```
data=[1,5,8,7,9,6,3,1,4,11,20]
a=0
for i in data :
    a=a+i
moy=a/len(data)
```

x **Déterminer** la valeur de i pour la première puis pour la dernière exécution de la boucle for.

Première boucle : i vaut 1

Dernière boucle : i vaut 20

x **Déterminer** la valeur de a à la fin de l'exécution de la boucle for.

a la fin de la boucle for a vaut 75

x **Renommer** cette variable afin de la rendre plus explicite.

On peut la nommer *somme*

x **Déterminer** la valeur de moy à la fin de l'exécution du programme.

Moy vaut 6,81818181

x **En déduire** le rôle de ce programme.

Ce programme calcule la moyenne des valeurs contenues dans la liste

Exercice 4 :

x **Écrire** un programme qui compte le nombre d'éléments supérieurs à 10 dans une liste d'entiers nommée data.

```
nb=0
for i in data:
    if i>10:
        nb+=1
print(nb)
```

Exercice 5 : Concaténation de deux listes

Soient les listes suivantes :

```
t1 = [31, 28, 31, 30]
t2 = ['Janvier ', ' Février ', 'Mars ', 'Avril ']
```

x **Écrire** un programme qui crée une nouvelle liste t3 contenant tous les éléments des deux listes t1 et t2 alternés, de telle manière que chaque nom de mois soit suivi du nombre de jours correspondant :

```
[ ' Janvier ', 31, ' Février ', 28, ' Mars ', 31, etc...].
```

```
t3=[]
for i in range(len(t1)):
    t3.append(t2[i])
    t3.append(t1[i])
```

Exercice 6 : Tri de données

Soit la liste suivante :

```
data=[1,5,8,7,9,6,3,1,4,11,20]
```

x **Écrire** un programme qui génère deux nouvelles listes nommées listePaire et listeImpaire. listePaire doit contenir seulement les nombres pairs de la liste initiale, et listeImpaire les nombres impairs.

```
listePaire=[]
listeImpaire=[]
for i in data:
    if i%2==0: #test si i paire
        listePaire.append(i)
    else:
        listeImpaire.append(i)
```

x **Modifier** le programme précédent afin de ranger les éléments de la liste data dans une autre liste resultat, en mettant les éléments pairs à gauche et les éléments impairs à droite.

```
listePaire=[]
listeImpaire=[]
for i in data:
    if i%2==0: #test si i paire
        listePaire.append(i)
    else:
        listeImpaire.append(i)
```

```
resultat= listePaire+ listeImpaire
```