

Exercices sur les listes

Exercice 1 :

Écrire un programme qui crée une liste suivante :

0	1	2	3	4
8	5	6	1	17

puis successivement :

- affiche la valeur de $L[4]$
- modifie la liste en remplaçant $L[1]$ par 17 et $L[3]$ par la somme des cases voisines $L[2]$ et $L[4]$
- affiche 12 fois la valeur du dernier terme de la liste

Exercice 2 : Échange de valeurs

Écrire un programme qui échange les valeurs de la première et de la dernière case de la liste suivante :

0	1	2	3
6	12	18	5

Exercice 3 : Parcours de liste

A l'aide de parcours successifs de la liste, effectuer les actions suivantes :

1. Affiche la liste $L = [8, 24, 48, 2, 16]$ en colonne :
2. Compte le nombre de multiples de 3 présents dans la liste $L = [8, 24, 48, 2, 16]$
3. Calcule la somme de toutes les valeurs paires de la liste $L = [8, 24, 48, 2, 16]$
4. Calcule le maximum et le minimum des éléments de la liste $L = [8, 24, 48, 2, 16]$
5. Créer un booléen VRAI si la moyenne arithmétique des valeurs de la liste $L = [8, 24, 48, 2, 16]$ est supérieure ou égale à 10.
6. Créer le produit de toutes les valeurs de la liste $L = [8, 24, 68, 2, 56]$ comprises dans l'intervalle $[50;70]$.
7. Affiche la liste $L = [8, 24, 48, 2, 16]$ à l'envers (sans créer de nouvelle liste)

Exercice 4 : Modifications de listes

Écrire un programme qui crée la liste d'entiers $L = [13, 27, 50, 42, 3]$ puis qui effectue successivement les actions suivantes :

1. Modifie la liste en augmentant de 1 la valeur de chaque élément de la liste
2. Modifie la liste par permutation circulaire
3. Modifie la liste par miroir

Exercice 5 : Création de listes

1. On dispose d'une liste $L = [8, 27, 42, 17, 4]$.
Créer une liste LC contenant les carrés des nombres de L.
2. On dispose maintenant de deux listes L1 et L2. Il s'agit de créer une liste :
 - une liste L3 identique à L1 dans laquelle on a supprimé la première valeur
 - une liste L4 constituée des valeurs L1 suivies des valeurs de L2

Exercice 6 : Symétrie de la liste

Écrire un programme qui affiche si la liste $L = [56, 1, 21, 16, 17]$ est symétrique (liste identique à la liste à l'envers).

Exercice 7 :

Écrire un programme qui renvoie la liste des premiers carrés d'entiers, en utilisant une fonction.

CORRIGE – Notre Dame de La Merci – Montpellier

Exercice 1 :

Écrire un programme qui crée une liste suivante :

0	1	2	3	4
8	5	6	1	17

puis successivement :

- affiche la valeur de L[4]
- modifie la liste en remplaçant L[1] par 17 et L[3] par la somme des cases voisines L[2] et L[4]
- affiche 12 fois la valeur du dernier terme de la liste

```
L = [8, 5, 6, 1, 17]
print (L[4])
L[1] = 17
L[3] = L[2] + L[4]
print (L)
for i in range (0, 12) :
    print (L[len(L)-1])
```

Exercice 2 : Échange de valeurs

Écrire un programme qui échange les valeurs de la première et de la dernière case de la liste suivante :

0	1	2	3
6	12	18	5

```
L = [6, 12, 18, 5]
aux = L[0]           # utilisation d'une variable auxiliaire
print (L)
L[0] = L[3]
print (L)
L[3] = aux
print (L)
```

AUTRE METHODE :

```
L = [6, 12, 18, 5]
print (L)
L[0],L[3] = L[3],L[0]
print (L)
```

→[5, 12, 18, 6]

Exercice 3 : Parcours de liste

A l'aide de parcours successifs de la liste, effectuer les actions suivantes :

1. Affiche la liste L = [8, 24, 48, 2, 16] en colonne

```
L = [8, 24, 48, 2, 16]
print(L)
long = len(L)
#affiche en colonne
for i in range (long) :
    print (i, " : ", L[i])
```

2. Compte le nombre de multiples de 3 présents dans la liste L = [8, 24, 48, 2, 16]

```
L = [8, 24, 48, 2, 16]
long = len(L)
```

```
#nombre de multiples de 3 présents dans L
mult3 = 0
for i in range (long) :
    if L[i] %3 == 0 :
        mult3 = mult3 + 1
        print(L[i])
print ("Nombre de multiples de 3 : ", mult3)
```

→ 24
48
Nombre de multiples de 3 : 2

3. Calcule la somme de toutes les valeurs paires de la liste $L = [8, 24, 48, 2, 16]$

```
L = [8, 24, 48, 2, 16]
long = len(L)
#somme des valeurs paires de L
somp = 0
for i in range (long) :
    if L[i] %2 == 0 :
        somp = somp + L[i]
print ("Somme des valeurs paires : ", somp)
```

→ Somme des valeurs paires : 98

4. Calcule le maximum et le minimum des éléments de la liste $L = [8, 24, 48, 2, 16]$

```
L = [8, 24, 48, 2, 16]
long = len(L)
#calcul du maximum et du minimum de L
max = L[0]
min = L[0]
for i in range (1,long) :
    if L[i] > max :
        max = L[i]
    else :
        if L[i] < min :
            min = L[i]
print ("Max = ", max, "et min = ", min)
```

→ Max = 48 et min = 2

5. Créer un booléen VRAI si la moyenne arithmétique des valeurs de la liste $L = [8, 24, 48, 2, 16]$ est supérieure ou égale à 10.

```
L = [8, 24, 48, 2, 16]
long = len(L)
#la moyenne arithmétique des valeurs de la liste est supérieure ou égale à 10 ?
somme = 0
for i in range (long) :
    somme += L[i]
moy = somme/long
if moy >= 10 :
    print ("La moyenne est supérieure ou égale à 10 : ", moy)
else :
    print ("La moyenne est inférieure à 10 : ", moy)
```

→ La moyenne est supérieure ou égale à 10 : 19.6

6. Créer le produit de toutes les valeurs de la liste $L = [8, 24, 68, 2, 56]$ comprises dans l'intervalle $[50; 70]$.

```
L = [8, 24, 68, 2, 56]
long = len(L)
#produit des valeurs de [50, 70]
produit = 1
for i in range(0, long) :
    if 50 <= L[i] <= 70 :
        produit = produit *L[i]
print ("produit des valeurs comprises entre 50 et 70 : ", produit)
```

→ produit des valeurs comprises entre 50 et 70 : 3808

7. Affiche la liste $L = [8, 24, 48, 2, 16]$ à l'envers (sans créer de nouvelle liste)

```
L = [8, 24, 48, 2, 16]
long = len(L)
#affiche à l'envers
print ("Affichage à l'envers :")
for i in range(long) :
    print (L[long-1-i], ', ', end=" ")
```

→ 16 , 2 , 48 , 24 , 8 ,

NB : avec **print** (L[long-1-i]) seulement, on obtient :

```
16
2
48
42
8
```

Exercice 4 : Modifications de listes

Écrire un programme qui crée la liste d'entiers $L = [13, 27, 50, 42, 3]$ puis qui effectue successivement les actions suivantes :

1. Modifie la liste en augmentant de 1 la valeur de chaque élément de la liste

```
L = [13, 27, 50, 42, 3]
long = len(L)
print(L)
#ajout de 1 à chaque valeur de L
for i in range(long) :
    L[i] += 1
print ("L modifiée par ajout de 1 : ")
print(L)
```

→ [13, 27, 50, 42, 3]
L modifiée par ajout de 1 :
[14, 28, 51, 43, 4]

2. Modifie la liste par permutation circulaire

```
L = [8, 24, 48, 2, 16]
long = len(L)
#permutation circulaire
aux = L[0]
for i in range (1, long) :           #for i in range(0,long-1) :
    L[i-1] = L[i]                     #    L[i] = L[i+1]
L[long-1] = aux
print ("apres permutation circulaire : ", L)

→ apres permutation circulaire : [24, 48, 2, 16, 8]
```

3. Modifie la liste par miroir

```
L = [8, 24, 48, 2, 16]           # L = [8, 24, 48, 2, 16,12]
long = len(L)
#liste miroir
for i in range (0, (long)//2) :
    aux = L[i]
    L[i] = L[long-i-1]
    L[long-i-1] = aux
print ("liste miroir : ", L)
```

Ou

```
L = [8, 24, 48, 2, 16]   # L = [8, 24, 48, 2, 16, 12]
long = len(L)
#liste miroir
for i in range (0, (long)//2) :
    L[i], L[long-i-1] = L[long-i-1], L[i]
print ("liste miroir : ", L)
```

Ou

```
L = [8, 24, 48, 2, 16]   # L = [8, 24, 48, 2, 16, 12]
long = len(L)
L.reverse()
print ("liste miroir : ", L)
```

→ liste miroir : [16, 2, 48, 24, 8]

Exercice 5 : Création de listes

1. On dispose d'une liste L = [8, 27, 42, 17, 4]. Créer une liste LC contenant les carrés des nombres de L.

```
L = [8, 27, 42, 17, 4]
long = len(L)
LC = []
#creation de listes
for i in range (long) :
    LC.append(L[i]**2)
print (LC)
```

→ [64, 729, 1764, 289, 16]

2. On dispose maintenant de deux listes L1 et L2. Il s'agit de créer une liste :

- une liste L3 identique à L1 dans laquelle on a supprimé la première valeur
- une liste L4 constituée des valeurs L1 suivies des valeurs de L2

```
L1 = [8.3, 27.2, 42, 17, 4.1]
print ("L1 = ", L1)
#L3 avec premiere valeur supprime
L3 = []
for i in range (1, len(L1)) :
    L3.append(L1[i])
print ("L3 = ", L3)

→ L1 = [8.3, 27.2, 42, 17, 4.1]
   L3 = [27.2, 42, 17, 4.1]
```

```
L1 = [8.3, 27.2, 42, 17, 4.1]
L2 = [15, 5.9, 59, 10.1, 2]
print ("L1 = ", L1, "et L2 = ", L2)
#L4 avec valeurs de L1 puis L2
L4 = []
for i in range (0, len(L1)) :
    L4.append(L1[i])
for i in range (0, len(L2)) :
    L4.append(L2[i])
print ("L4 = ", L4)

→ L1 = [8.3, 27.2, 42, 17, 4.1] et L2 = [15, 5.9, 59, 10.1, 2]
   L4 = [8.3, 27.2, 42, 17, 4.1, 15, 5.9, 59, 10.1, 2]
```

Exercice 6 : Symétrie de la liste

Écrire un programme qui affiche si la liste L = [56, 1, 21, 16, 17] est symétrique (liste identique à la liste à l'envers).

```
L = [56, 1, 21, 16, 17]
long = len(L)
#la liste est-elle symetrique ?
sym="vrai"
for i in range (0, long//2) :
    if L[i] != L[len(L)-1-i] :
        sym="false"
if sym=="vrai":
    print("La liste est symetrique")
else:
    print("La liste n'est pas symetrique")

→ La liste n'est pas symetrique
```

Exercice 7 :

Écrire un programme qui renvoie la liste des premiers carrés d'entiers, en utilisant une fonction.

```
def nb_carre(a):
    liste=[]
    for i in range(1,a+1):
        liste.append(i**2)
    return liste

valeur=int(input("Veuillez saisir un rang:"))
liste_carre=nb_carre(valeur)
print(liste_carre)

→ pour n = 10 : [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```