

## Exercices sur les listes

### Exercice 1A.1 :

Écrire un programme qui crée une liste suivante :

0	1	2	3	4
7	4	1	8	5

puis successivement :

- affiche la valeur de L[4]
- modifie la liste en remplaçant L[1] par 23 et L[3] par la somme des cases voisines L[2] et L[4]
- affiche 6 fois la valeur du dernier terme de la liste

### Exercice 1A.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
91	84	77	70

### Exercice 1A.3 : Parcours de liste

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

1. Afficher la liste  $L = [12, 44, 38, 72, 19, 93]$  en colonne :
2. Compter le nombre de multiples de 3 présents dans la liste  $L = [44, 12, 93, 72, 19, 38]$
3. Calculer la somme de toutes les valeurs paires de la liste  $L = [44, 12, 93, 72, 19, 38]$
4. Calculer le maximum et le minimum des éléments de la liste  $L = [44, 12, 93, 72, 19, 38]$
5. Créer un booléen VRAI si la moyenne arithmétique des valeurs de la liste  $L = [44, 12, 93, 72, 19, 38]$  est supérieure ou égale à 40.
6. Créer le produit de toutes les valeurs de la liste  $L = [44, 12, 93, 72, 19, 38]$  comprises dans l'intervalle  $[30;80]$ .
7. Afficher la liste  $L = [44, 12, 93, 72, 19, 38]$  à l'envers (sans créer de nouvelle liste)

### Exercice 1A.4 : Modifications de listes

Écrire un programme qui crée la liste d'entiers  $L = [14, 49, 27, 38, 9]$  puis qui effectue successivement les actions suivantes :

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

### Exercice 1A.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 1A.6 : Symétrie de la liste

Écrire un programme qui affiche si la liste  $L = [34, 22, 54, 22, 34]$  est symétrique (liste identique à la liste à l'envers).

### Exercice 1A.7 :

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

**CORRIGE – Notre Dame de La Merci – Montpellier – M. Quet**

**Exercice 1A.1 :**

Écrire un programme qui crée une liste suivante :

0	1	2	3	4
7	4	1	8	5

puis successivement :

- affiche la valeur de L[4]
- modifie la liste en remplaçant L[1] par 23 et L[3] par la somme des cases voisines L[2] et L[4]
- affiche 6 fois la valeur du dernier terme de la liste

```
L = [7,4,1,8,5]
print(L)
print (L[4])
L[1] = 23
L[3] = L[2] + L[4]
print (L)
for i in range (6) :
    print (L[len(L)-1])
```

On obtient :

```
[7, 4, 1, 8, 5]
5
[7, 23, 1, 6, 5]
5
5
5
5
5
5
5
```



**Exercice 1A.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
91	84	77	70

```
L = [91,84,77,70]
print(L)
L[0],L[3] = L[3],L[0]
print(L)
```

On obtient :

```
[91, 84, 77, 70]
[70, 84, 77, 91]
```



**Exercice 1A.3 : Parcours de liste**

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

1. Afficher la liste L = [44, 12, 93, 72, 19, 38] en colonne

```
L = [44, 12, 93, 72, 19, 38]
print(L)
for i in range (len(L)):
    print (i, " : ", L[i])
```

On obtient :

```
0 : 44
1 : 12
2 : 93
3 : 72
4 : 19
5 : 38
```

2. Compter le nombre de multiples de 3 présents dans la liste  $L = [44, 12, 93, 72, 19, 38]$

```
L = [44, 12, 93, 72, 19, 38]
nb_mult3 = 0
for i in range (len(L)) :
    if L[i] %3 == 0 :
        nb_mult3 += 1
        print(L[i])
print ("Nombre de multiples de 3 : ", nb_mult3)
```

On obtient :

```
12
93
72
Nombre de multiples de 3 : 3
```

3. Calculer la somme de toutes les valeurs paires de la liste  $L = [44, 12, 93, 72, 19, 38]$

```
L = [44, 12, 93, 72, 19, 38]
som_pair = 0
for i in range (len(L)) :
    if L[i] %2 == 0 :
        som_pair += L[i]
print ("Somme des valeurs paires : ", som_pair)
```

On obtient :

```
Somme des valeurs paires : 166
```

4. Calculer le maximum et le minimum des éléments de la liste  $L = [44, 12, 93, 72, 19, 38]$

```
L = [44, 12, 93, 72, 19, 38]
max = L[0]
min = L[0]
for i in range (1,len(L)) :
    if L[i] > max :
        max = L[i]
    elif L[i] < min :
        min = L[i]
print ("Max = ", max, "et min = ", min)
```

On obtient :

```
Max = 93 et min = 12
```

5. Créer un booléen VRAI si la moyenne arithmétique des valeurs de la liste  $L = [44, 12, 93, 72, 19, 38]$  est supérieure ou égale à 40.

```
L = [44, 12, 93, 72, 19, 38]
somme = 0
for i in range (len(L)):
    somme += L[i]
moy = somme/len(L)
if moy >= 40 :
    print ("VRAI: La moyenne est supérieure ou égale à 40 : " ,moy)
```

```
else :
    print ("FAUX: La moyenne est inférieure à 40 : " ,moy)
```

On obtient :

VRAI: La moyenne est supérieure ou égale à 40 : 46.333333333333336

6. Créer le produit de toutes les valeurs de la liste  $L = [44, 12, 93, 72, 19, 38]$  comprises dans l'intervalle  $[30;80]$ .

```
L = [44, 12, 93, 72, 19, 38]
produit = 1
for i in range (0, len(L)) :
    if 30 <= L[i] <= 80 :
        produit = produit *L[i]
print ("Produit des valeurs comprises entre 30 et 80 : " , produit)
```

On obtient :

Produit des valeurs comprises entre 30 et 80 : 120384

7. Afficher la liste  $L = [44, 12, 93, 72, 19, 38]$  à l'envers (sans créer de nouvelle liste)

```
L = [44, 12, 93, 72, 19, 38]
print(L)
for i in range (len(L)//2) :
    L[i], L[len(L)-i-1] = L[len(L)-i-1], L[i]
print(L)
```

On obtient :

```
[44, 12, 93, 72, 19, 38]
[38, 19, 72, 93, 12, 44]
```



### **Exercice 1A.4 : Modifications de listes**

Écrire un programme qui crée la liste d'entiers  $L = [14, 49, 27, 38, 9]$  puis qui effectue successivement les actions suivantes :

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

```
L = [14, 49, 27, 38, 9]
print(L)
for i in range (len(L)) :
    L[i] += 1
print(L)
```

On obtient :

```
[14, 49, 27, 38, 9]
[15, 50, 28, 39, 10]
```

2. Modifier la liste par permutation circulaire

```
L = [14, 49, 27, 38, 9]
print (L)
aux = L[0]
for i in range (1, len(L)) :
    L[i-1] = L[i]      #
L[len(L)-1] = aux
print (L)
```

On obtient :

[14, 49, 27, 38, 9]

[49, 27, 38, 9, 14]

**3. Modifier la liste par miroir**

L = [14, 49, 27, 38, 9]

print (L)

for i in range (0, (len(L))/2) :

    L[i], L[len(L)-i-1] = L[len(L)-i-1], L[i]

print (L)

**Ou**

L = [14, 49, 27, 38, 9]

print (L)

L.reverse()

print (L)

On obtient :

[14, 49, 27, 38, 9]

[9, 38, 27, 49, 14]



**Exercice 1A.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]

print(L)

LC = []

for i in range (len(L)) :

    LC.append(L[i]\*\*2)

print(LC)

On obtient :

[8, 27, 42, 17, 4]

[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

L1 = [8.3, 27.2, 42, 17, 4.1]

print (L1)

L3 = []

for i in range (1, len(L1)) :

    L3.append(L1[i])

print (L3)

**Ou**

L1 = [8.3, 27.2, 42, 17, 4.1]

print (L1)

L3 = L1

del L3[0]

print (L3)

On obtient :

[8.3, 27.2, 42, 17, 4.1]

[27.2, 42, 17, 4.1]



- une liste L4 constituée des valeurs L1 suivies des valeurs de L2

```
L1 = [8.3, 27.2, 42, 17, 4.1]
L2 = [15, 5.9, 59, 10.1, 2]
print (L1)
print (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)
```

**Ou**

```
L1 = [8.3, 27.2, 42, 17, 4.1]
L2 = [15, 5.9, 59, 10.1, 2]
print (L1)
print (L2)
L4 = L1+L2
print ("L4 = ", L4)
```

On obtient :

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



### **Exercice 1A.6 : Symétrie de la liste**

Écrire un programme qui affiche si la liste  $L = [34, 22, 54, 22, 34]$  est symétrique (liste identique à la liste à l'envers).

```
L = [34, 22, 54, 22, 34]
sym="vrai"
for i in range (0, len(L)//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")
```

On obtient :

La liste est symetrique



### **Exercice 1A.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)
```

On obtient :

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]