

### EXERCICE 1 - ●○○○ - LISTE INVERSÉE

Écrire une fonction qui prend en argument d'entrée une liste et qui renvoie la liste obtenue en inversant l'ordre des éléments.

```
1 def listeinversee(L):
2     M=[]
3     for k in L:
4         M=[k]+M #permet d'ajouter l'élément par la gauche
5     return M
```

#### Petite remarque

La commande `L[::-1]` permet d'obtenir la liste inversée, mais il s'agit ici de ne pas l'utiliser.

### EXERCICE 2 - ●○○○ - SUPPRESSION

Écrire une fonction qui prend en arguments d'entrée une liste et un élément  $x$ , puis qui renvoie la nouvelle liste obtenue après suppression de toutes les occurrences de  $x$  dans cette liste.

```
1 def suppression(L,x):
2     M=[k for k in L if k!=x] #nouvelle liste avec les éléments différents de x
3     return M
4
5 def suppression_bis(L,x):
6     M=[k for k in L if k!=x] #nouvelle liste avec les éléments différents de x
7     L=M #on modifie la liste initiale
8     return L
```

#### Petite remarque

Le programme "naïf" consistant à supprimer `L[k]` lorsqu'il est égal à  $x$  pose souci... Et autant profiter des spécificités de Python !

### EXERCICE 3 - ●○○○ - INSERTION

Écrire une fonction qui prend en arguments d'entrées une liste, un indice  $i$  et un élément  $x$ , puis qui renvoie la nouvelle liste obtenue après insertion de l'élément  $x$  à l'indice  $i$ , en décalant le reste vers la droite.

```
1 def insertion(L,i,x):
2     L1=L[0:i]
3     L2=L[i:len(L)]
4     L=L1+[x]+L2 #on modifie la liste initiale
5     return L
```

#### Petite remarque

La commande `L.insert(i,x)` permet de le faire, mais elle n'est pas à connaître.

### EXERCICE 4 - ●○○○ OCCURRENCES...

1. Écrire une fonction prenant en arguments d'entrée une liste  $L$  et un élément  $x$ , puis renvoyant en sortie le nombre d'occurrences de  $x$  dans  $L$ .

```
1 def occurence(L,x):
2     c=0
3     for e in L:
4         if e==x:
5             c=c+1
6     return c
```

2. Écrire une fonction prenant en arguments d'entrée une liste  $L$  et un élément  $x$ , puis renvoyant en sortie le rang de la première occurrence de  $x$  dans  $L$  si la liste  $L$  contient l'élément  $x$ , et `False` sinon.

```
1 def occ(L,x):
2     for k in range(0,len(L)):
3         if L[k]==x:
4             return k
5     return False
```

### EXERCICE 5 - ●○○○ MAXIMUM

1. Sans utiliser la commande `max`, écrire une fonction prenant en argument d'entrée une liste de réels `L` puis renvoyant en sortie le maximum de `L` ainsi que son nombre d'occurrences.

```
1 def monmax(L):
2     maxi=L[0]
3     c=0
4     for x in L:
5         if x>maxi:
6             maxi=x
7             c=1
8         elif x==maxi:
9             c=c+1
10    return maxi , c
```

2. Sans utiliser la commande `max`, écrire une fonction prenant en argument d'entrée une liste de réels `L` puis renvoyant en sortie le maximum de `L` ainsi que le rang de sa première occurrence.

```
1 def monmax(L):
2     maxi=L[0]
3     rang=0
4     for k in range(1, len(L)):
5         if L[k]>maxi:
6             maxi=L[k]
7             rang=k
8     return maxi , rang
```

3. Sans utiliser la commande `max`, écrire une fonction prenant en argument d'entrée une liste de réels `L` puis renvoyant en sortie les deux plus grands nombres (éventuellement égaux) de cette liste.

```
1 def deux_plus_grands(L):
2     if L[0]<L[1]:
3         maxi1 , maxi2=L[1] , L[0]
4     else:
5         maxi1 , maxi2=L[0] , L[1]
6     for k in L[2: len(L)]:
7         if k>maxi1 and k>maxi2:
8             maxi1 , maxi2=k , maxi1
9         elif maxi2<k<=maxi1:
10            maxi2=k
11    return maxi2
```

4. Sans utiliser la commande `max`, écrire une fonction prenant en argument d'entrée une liste de réels `L` dont au moins deux sont distincts puis renvoyant en sortie le second plus grand maximum (distinct du premier) de cette liste.

```
1 def second_max(L):
2     k=1
3     while L[k]==L[0]:
4         k=k+1
5     if L[0]<L[k]:
6         maxi1 , maxi2=L[k] , L[0]
7     elif L[0]>L[k]:
8         maxi1 , maxi2=L[0] , L[k]
9     for x in L[2: len(L)]:
10        if x>maxi1 and x>maxi2:
11            maxi1 , maxi2=x , maxi1
12        elif maxi2<x<maxi1:
13            maxi2=x
14    return maxi2
```

## EXERCICE 6 - ●●○○ LES PLUS PROCHEs

Écrire une fonction prenant en argument d'entrée une liste de réels `L` (contenant au moins deux éléments) puis renvoyant en sortie les deux éléments les plus proches.

```

1 def proches(L):
2     ecart=abs(L[1]-L[0])
3     a,b=L[0],L[1]
4     for i in range(0,len(L)):
5         for j in range(i+1,len(L)):
6             if abs(L[i]-L[j])<ecart:
7                 ecart=abs(L[i]-L[j])
8                 a,b=L[i],L[j]
9     return a,b

```

**Petite remarque**

En mettant  
 ◀ **if 0<abs(L[i]-L[j])<ecart**  
 en ligne 6, on obtient les deux  
 plus proches distincts.

**EXERCICE 7 - ●●●○ - ORDONNER UNE LISTE...**

Écrire une fonction qui prend en arguments d'entrée une liste de réels et qui renvoie la liste triée dans l'ordre croissant.

**Petite remarque**

◀ La commande **L.sort()** permet  
 d'ordonner une liste, mais il s'agit  
 ici de ne pas l'utiliser.

```

1 def listeordonnee(L):
2     for k in range(1,len(L)):
3         for j in range(0,k):
4             if L[j]>L[k]:
5                 L[j],L[k]=L[k],L[j]
6     return L

```