

```
# ===== Entrainement 1 =====  
def somme(L):  
    S = 0  
  
    for x in L:  
        S += x  
  
    return S
```

```
# ===== Entrainement 2 =====  
def suite2(n):  
    L = [1.]  
  
    for k in range(n):  
        u = 0  
        for i in range(k + 1):  
            u += L[i] / (k + 1 - i)  
        L.append(u)  
  
    return L[-1]
```

```
# ===== Entrainement 3 =====  
def indice_max(L):  
  
    record = -1  
  
    for i in range(len(L)):  
        if L[i] > record:  
            ind_record = i  
            record = L[i]  
  
    return ind_record
```

```
# ===== Entrainement 4 =====  
def alterne(L, M):  
  
    P = []  
    for i in range(len(L)):  
        P.append(L[i])  
        P.append(M[i])  
  
    return P
```

```
# ===== Entrainement 5 =====  
def images(f, L):  
  
    M = []  
  
    for x in L:  
        M.append(f(x))  
  
    return M  
  
def appliquer(f, L):  
  
    for i in range(len(L)):  
        L[i] = f(L[i])
```

```

# ===== Entrainement 6 =====
def est_croissant(L):
    i = 1
    n = len(L)

    while i < n and L[i - 1] <= L[i]:
        i += 1

    return i == n

def est_croissant_for(L):
    for i in range(1, len(L)):
        if L[i - 1] > L[i]:
            return False

    return True

```

```

# ===== Entrainement 7 =====
def supprime(L, x):
    i = 0

    while i < len(L):
        if L[i] == x:
            del L[i]
        else:
            i += 1

# Une version naïve avec for
def supprime_faux(L, x):
    for k in range(len(L)):
        if L[k] == x:
            del L[k]

# Ce code ne fonctionne pas car on change les indices
# et l'itérateur, lui, n'est pas modifié.
# On obtient un "list index out of range"

# On peut utiliser une boucle for à condition de parcourir la liste
# à partir de la fin :
def supprime_for(L, x):
    for k in range(len(L) - 1, -1, -1):
        if L[k] == x:
            del L[k]

# Version avec retour de valeur et sans effet de bord
def sans_valeur(L, x):
    return [a for a in L if a != x]

```