

Corrigé sujet **06** - Année : 2023

Sujet 06 - 2023 [↓](#)

Exercice 1

```
1 def recherche(elt, tab):
2     for i in range(len(tab)-1, -1, -1):
3         if elt==tab[i]:
4             return i
5     return len(tab)
```



Commentaires

Le sujet demande de rechercher la **dernière** occurrence, la correction proposée ici parcourt la liste à l'envers et renvoie la première occurrence rencontrée. Il faut donc dans ce cas savoir à écrire un parcours à l'envers à l'aide de `range`. On peut faire aussi parcourir dans le sens normal jusqu'à la fin et mettre la jour l'indice à chaque fois qu'on rencontre la valeur:

```
def recherche(tab, n):
    indice_solution = len(tab)
    for i in range(len(tab)):
        if tab[i] == n:
            indice_solution = i
    return indice_solution
```

Exercice 2

```
1 from math import sqrt # import de la fonction racine carree
2
3 def distance(point1, point2):
4     """ Calcule et renvoie la distance entre deux points. """
5     return sqrt((point1[0]-point2[0])**2 + (point1[1]-point2[1])**2) # ①
6
7 def plus_courte_distance(tab, depart):
8     """ Renvoie le point du tableau tab se trouvant a la plus
9     courte distance du point depart. """
10    point = tab[0]
11    min_dist = distance(point, depart) # ②
12    for i in range(1, len(tab)): # ③
13        if distance(tab[i], depart) < min_dist:
14            point = tab[i]
15            min_dist = distance(tab[i], depart)
16    return point
```

1. Un `point` est un tuple `(abscisse, ordonne)` donc `point[0]` contient l'abscisse et `point[1]` l'ordonnée.
2. On initialise le minimum à la distance entre le point de départ et le premier point de la liste (celui d'indice 0)
3. Algorithme classique de recherche du minimum