

# Corrigé sujet 36 - Année : 2022

[Sujet 36 - 2022 ↴](#)

## 1. 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:

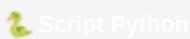


```

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

```

## 2. 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 assert distance((1, 0), (5, 3)) == 5.0, "erreur de calcul"
8

```

```
9 def plus_courte_distance(tab, depart):
10     """ Renvoie le point du tableau tab se trouvant à la plus
11     courte distance du point départ."""
12     point = tab[0]
13     min_dist = distance(point, depart) #(2)
14     for i in range (1, len(tab)): #(3)
15         if distance(tab[i], depart) < min_dist:
16             point = tab[i]
17             min_dist = distance(tab[i], depart)
18     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

### Attention

1. L'exercice demande aussi d'ajouter une ou des préconditions à la fonction `distance`. Les points ayant des coordonnées entières on peut proposer : `assert type(point[0])==int and type(point[1])==int`. On pourrait aussi s'assurer que `point` est un tuple de longueur 2.
2. On doit être très prudent en utilisant les tests d'égalité avec des flottants tels que celui de la ligne 7