

Nom :	Prénom :	Classe :
-------	----------	----------

NSI 1re — Algorithme de tri par sélection

Objectifs :

- Savoir **reconnaître** un algorithme de tri par sélection (d'un tableau indexé).
- Savoir en expliquer **le fonctionnement**, les principes.
- Savoir **écrire un algorithme de tri**.

Introduction

Dans un cours précédent, nous avons vu qu'il était possible de trier les éléments d'un tableau indexé selon une méthode — un algorithme — appelé algorithme de tri *par insertion*¹.

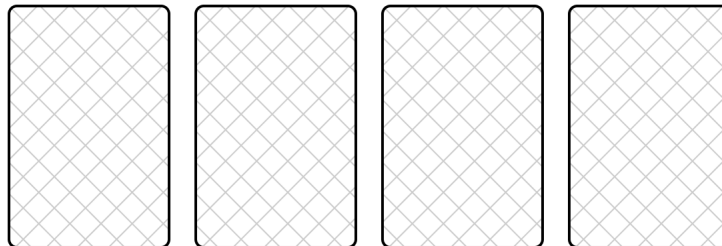
Tri par sélection

Le tri par sélection consiste à chercher à chaque étape le plus petit élément restant, et à le mettre à sa place au début du tableau.

Déroulé visuel

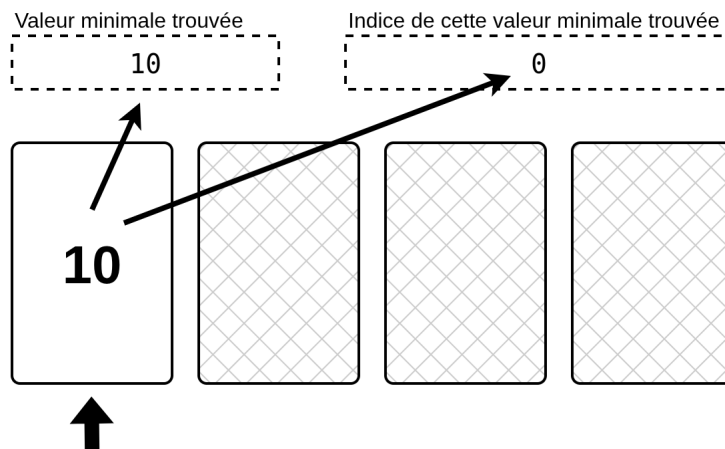
Voici un exemple de déroulé visuel, où l'on effectue un tri par sélection (dans l'ordre croissant) sur 4 cartes :

La première carte à gauche est la première **carte de référence**.



étape 1

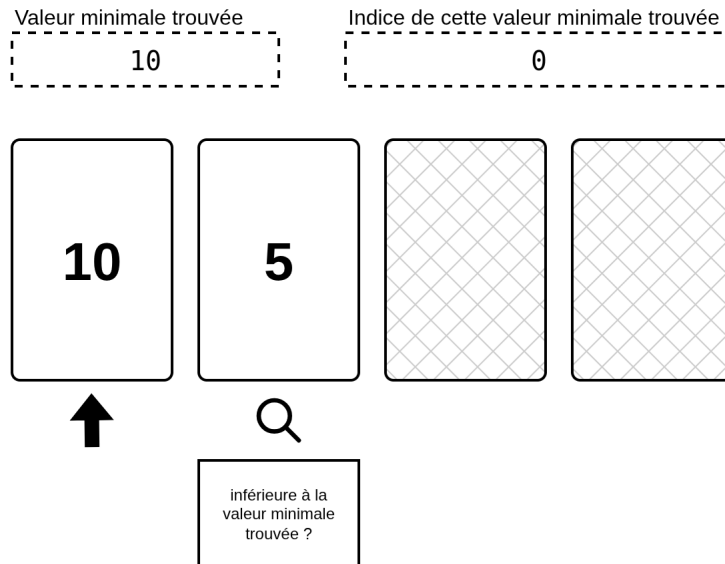
On met de côté sa valeur et son indice.



¹ Voir le cours « Algorithme de tri par insertion ».

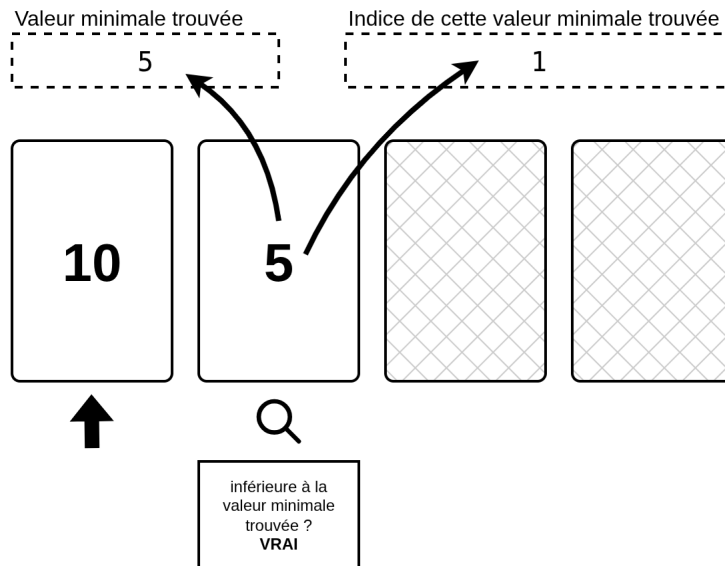
étape 2

Puis on cherche une carte de valeur plus faible, parmi les cartes situées à sa droite.



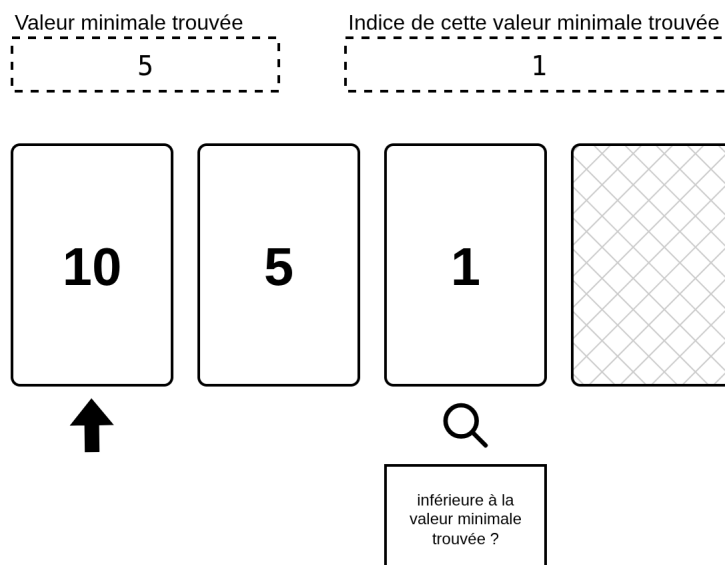
étape 3

Dès qu'une carte plus faible est trouvée, on met de côté sa valeur et son indice.

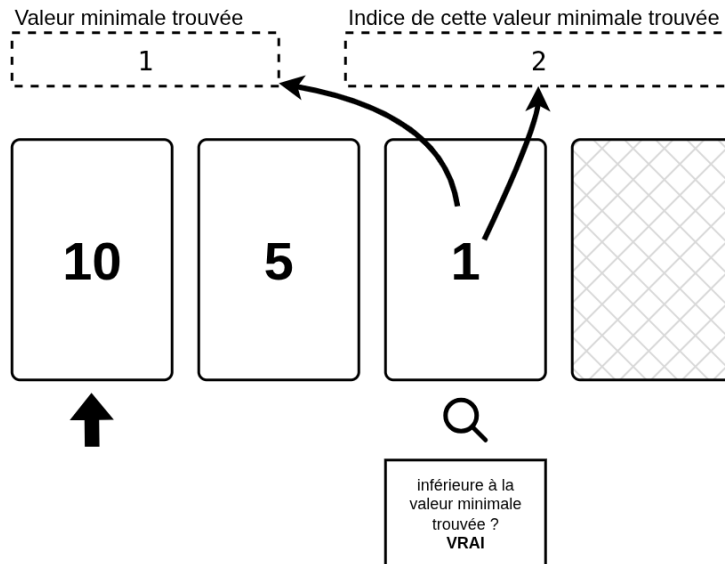


étape 4

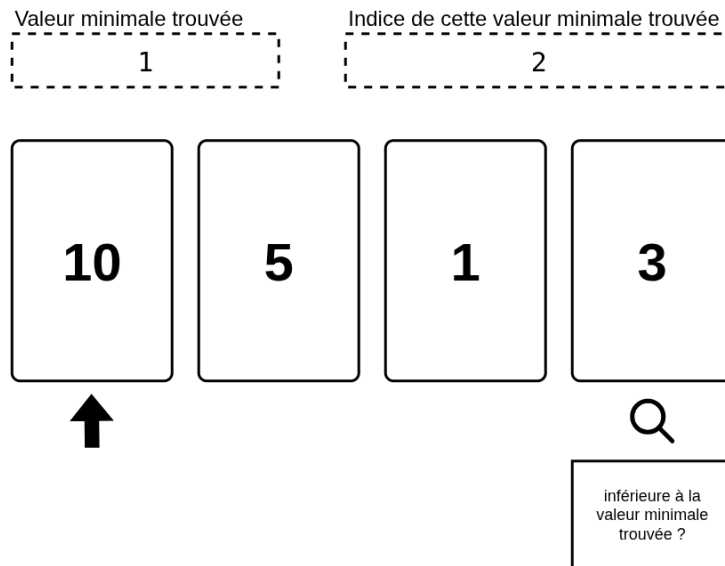
...



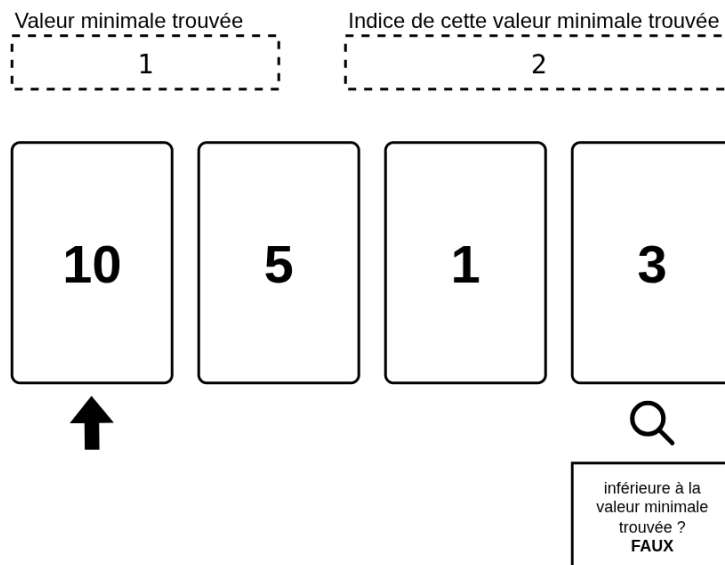
étape 5



étape 6



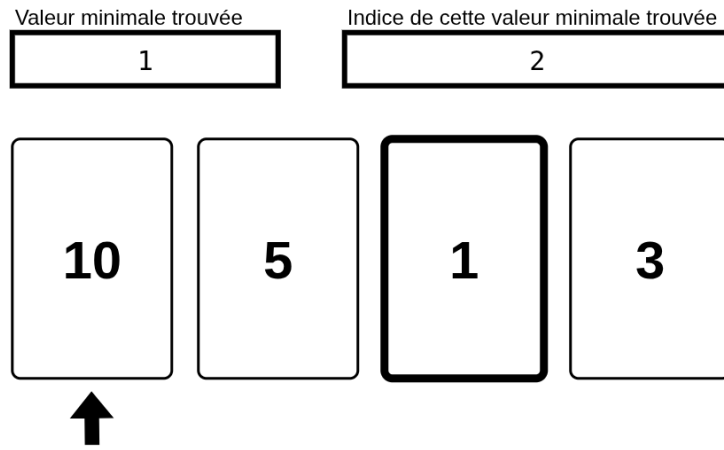
étape 7



étape 8

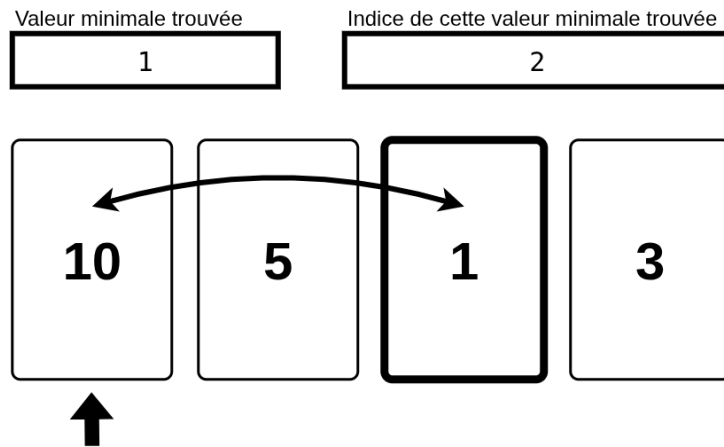
Les cartes à droite de la **carte de référence** ont été parcourues.

On connaît l'indice de la carte la plus faible.



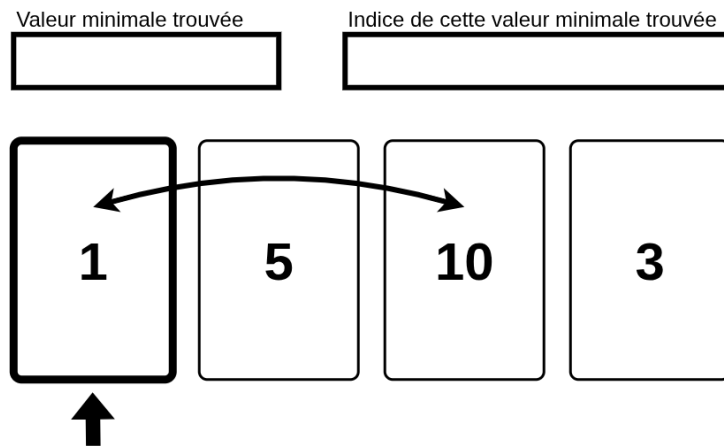
étape 9

Grâce à l'indice de la carte la plus faible, on échange celle-ci avec notre carte de référence.



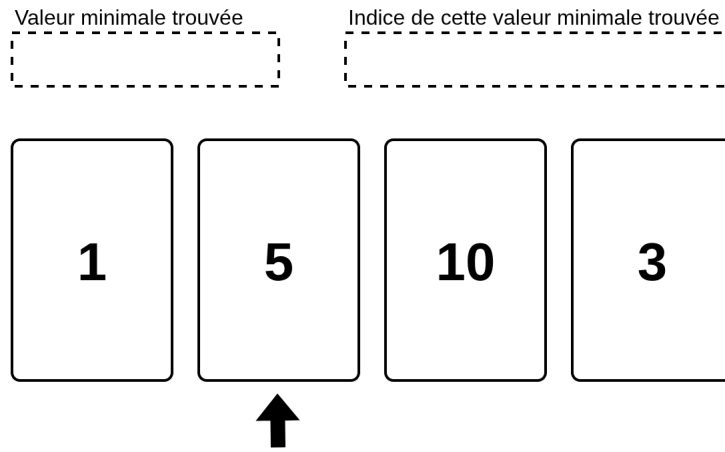
étape 10

Cartes échangées.



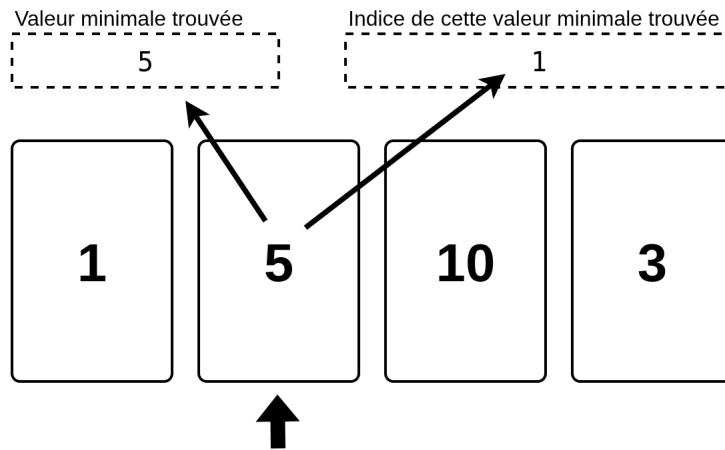
étape 11

La deuxième carte devient la nouvelle **carte de référence**.



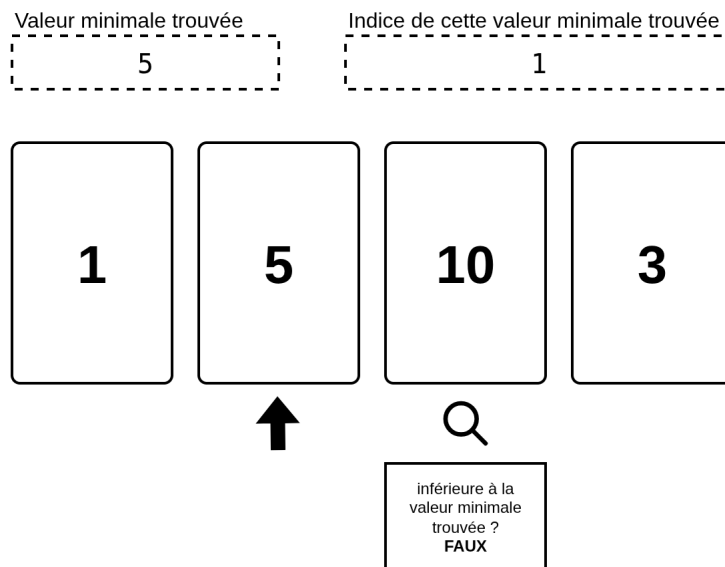
étape 12

On met de côté sa valeur et son indice.



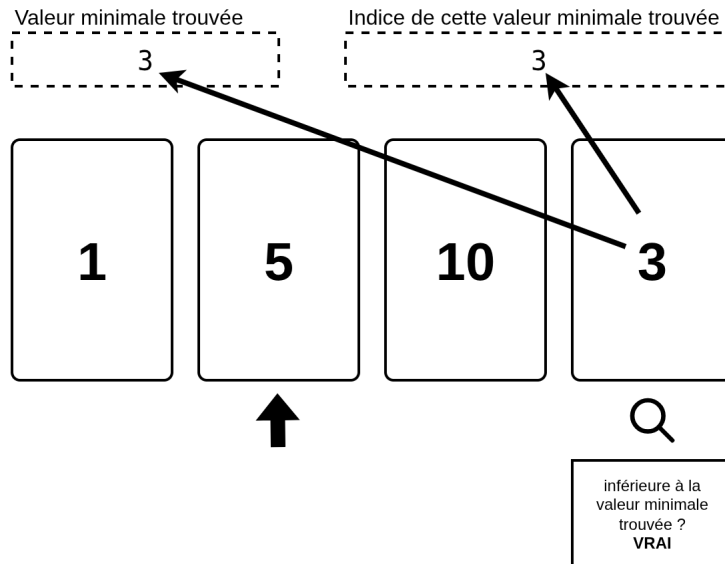
étape 13

Puis on cherche une carte de valeur plus faible, parmi les cartes situées à sa droite.



étape 14

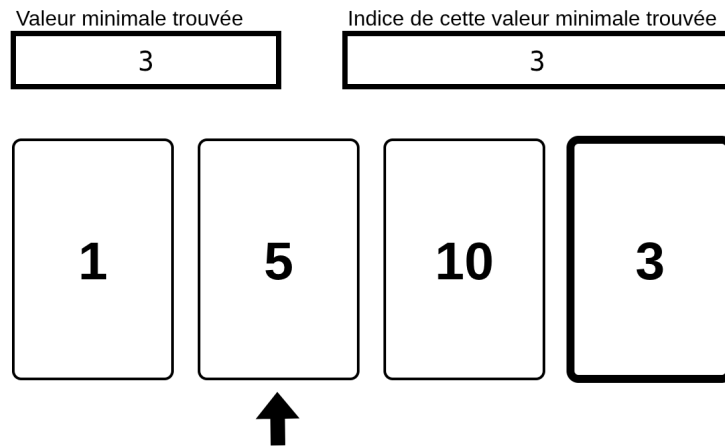
Dès qu'une carte plus faible est trouvée, on met de côté sa valeur et son indice.



étape 15

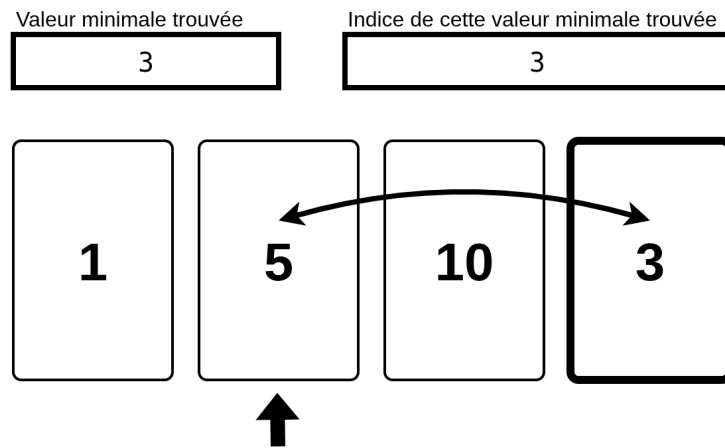
Les cartes à droite de la **carte de référence** ont été parcourues.

On connaît l'indice de la carte la plus faible.



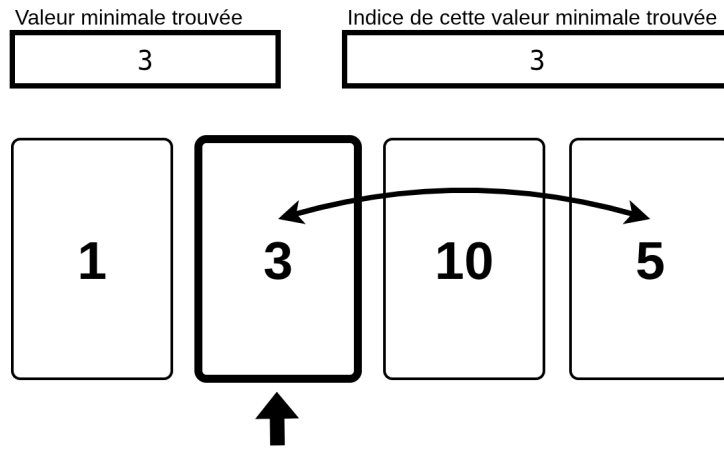
étape 16

Grâce à l'indice de la carte la plus faible, on échange celle-ci avec notre carte de référence.



étape 17

Cartes échangées.



étape 18

La troisième carte
devient la nouvelle
carte de référence ...

...

Etc.

Une fois la logique comprise grâce à ce « déroulé visuel », nous allons pouvoir représenter les étapes d'un tri par sélection sous la forme d'un tableau, page suivante.



Déroulé détaillé

Il est également possible de représenter les étapes d'un tri par sélection (ordre croissant) sous la forme d'un déroulé détaillé :

	Carte	Carte	Carte	Carte	Valeur min.	Indice
Départ	10	5	1	3		
Référence	10	5	1	3	10	0
Valeur plus faible ?	10	5	1	3	5	1
Valeur plus faible ?	10	5	1	3	1	2
Valeur plus faible ?	10	5	1	3	1	2
Échange	1	5	10	3		
Référence	1	5	10	3	5	1
Valeur plus faible ?	1	5	10	3	5	1
Valeur plus faible ?	1	5	10	3	3	3
Échange	1	3	10	5		
Référence	1	3	10	5	10	2
Valeur plus faible ?	1	3	10	5	5	3
Échange	1	3	5	10		
Fin	1	3	5	10		

► Exercice 1 — Enquêter

À vous de jouer, complétez les étapes d'un tri par sélection (ordre croissant) des éléments 80, 7 et 48, sous la forme d'un *déroulé détaillé* :

	Carte	Carte	Carte	Valeur min.	Indice
Départ	80	7	48		
Référence	80	7	48	80	0
Valeur plus faible ?	80	7	48	7	1
Valeur plus faible ?	80	7	48	7	1
Échange	7	80	48		
Référence	7	80	48	80	1
Valeur plus faible ?	7	80	48	48	2
Échange	7	48	80		
Fin	7	48	80		

Implémentation en pseudo-code

Comme tout algorithme, le tri par sélection peut être implémenté dans n'importe quel langage de programmation. Une première approche pourrait être d'analyser une implémentation possible en *pseudo-code*²:

Tri par sélection dans l'ordre croissant d'un tableau indexé :

```
1  Pour i de 0 à Taille(tableau) - 1
2      valeur_min = tableau[i]
3      indice_min = i
4      Pour j de i + 1 à Taille(tableau)
5          valeur_observée = tableau[j]
6          Si valeur_observée < valeur_min
7              indice_min = j
8              valeur_min = tableau[j]
9      tableau[indice_min] = tableau[i]
10     tableau[i] = valeur_min
```

² Un pseudo-code, c'est une manière simple d'écrire les étapes d'un programme sans utiliser un vrai langage de programmation.

Implémentation en Python

► Exercice 2 — Enquêter

Convertissez le pseudo-code du tri par sélection dans l'ordre croissant en langage Python :

```
for i in range(len(tableau) - 1):
    # Référence
    valeur_min = tableau[i]
    indice_min = i
    # Recherche d'une valeur plus faible
    # parmi les éléments situés à droite
    for j in range(i + 1, len(tableau)):
        # Valeur de cette carte
        valeur_observede = tableau[j]
        # La valeur de cette carte est plus faible ?
        if valeur_observede < valeur_min:
            # Oui ? Donc on mémorise
            indice_min = j
            valeur_min = valeur_observede
    # Fin de la recherche.
    # On passe à l'échange
    tableau[indice_min] = tableau[i]
    tableau[i] = valeur_min
```

Pour aller plus loin

N'hésitez pas à consulter cette petite vidéo expliquant le tri par sélection :



https://fmr.tf/s/le_tri_par_selection.mp4