

Série 6 (Fonctions - Tableau - Recherche - Tri)

1. Soit le tableau suivant contenant des valeurs triées par ordre croissant (peu importe les valeurs).

```
int[] tab = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
```

Veuillez écrire une fonction `indexOf(...)` permettant de savoir si une valeur se trouve dans ce tableau.

Ex :

```
System.out.println(indexOf(tab, 11)); // -1 (Pour indiquer que la valeur n'a pas été trouvée)
System.out.println(indexOf(tab, 1)); // 0
System.out.println(indexOf(tab, 7)); // 6
```

Indications : On commencera la recherche au milieu du tableau. Si la valeur recherchée est plus petite que la valeur au milieu du tableau on continuera la recherche au milieu du demi tableau de gauche. Si la valeur recherchée est plus grande que la valeur au milieu du tableau on continuera la recherche au milieu du demi tableau de droite (et ainsi de suite)

2. Veuillez récupérer le projet `ProjetRechercheElementTableau`

Combien de comparaisons faudra-t-il faire au maximum pour trouver l'index d'un élément dans un tableau de 300'000 éléments triés par ordre croissant ?

3. Veuillez récupérer le projet `ProjetAlgorithmesDeTri` qui contient une implémentation des algorithmes suivants :
 - Tri à bulles
 - Tri par sélection
 - Tri par insertion
 - Tri par tas
 - Tri par fusion
 - Tri rapide

Veuillez écrire le code permettant de classer ces différents algorithmes selon leur efficacité (en temps)

Indications : Chaque algorithme devra trier 10 tableaux de 50'000 valeurs aléatoires comprises entre 1 et 100'000.

On calculera alors le temps moyen de tri de chaque algorithmes.

4. Ecrire une fonction permettant d'insérer un élément dans un tableau de 20 cases de sorte qu'après chaque insertion les valeurs du tableau soient toujours dans ordre croissant.

Exemple :

insertion de 5 => {5}

insertion de 2 => {2, 5}

insertion de 7 => {2, 5, 7}

insertion de 1 => {1, 2, 5, 7}

insertion de 3 => {1, 2, 3, 4, 7}

... et ainsi de suite

5. Ecrire une fonction permettant de trier un tableau de chaînes de caractères (String) par ordre alphabétique.

{"pomme", "fraise", "ananas", "citron", "banane"} => {"ananas", "banane", "citron", "fraise", "pomme"}

6. Soit les personnes suivantes avec leur âges :

Paul	23
Simone	45
Louis	24
Anaïs	16

Ecrire un programme (avec des fonctions) permettant de trier les personnes d'après leurs prénoms,

Anaïs	16
Louis	24
Paul	23
Simone	45

ou d'après leur âges.

Anaïs	16
Paul	23
Louis	24
Simone	45