

Laboratoire Préparation à l'intra #1

Objectifs : Être capable de :

- Utiliser les structures de contrôle;
- Exploiter les tableaux;
- Définir et utiliser des sous-programmes (méthodes);
- Pratiquer les algorithmes de tris et de fouille.

Essayez d'abord de faire les exercices sans documentation et sans ordinateur.

Exercice 1 :

Un tableau de nombre entiers est sans doublon si aucune des valeurs du tableau n'est présente plus d'une fois.

Proposez une méthode **estSansDoublon()** qui reçoit un tableau de nombre entiers et qui retourne **true** si le tableau est sans doublon et **false** sinon.

Exercice 2 :

Proposez une méthode **getNombreValeursPresentes()** qui reçoit un tableau **t** de nombre entiers et un tableau **v** de nombres entiers et qui retourne le nombre de valeurs du tableau **v** qui sont présentes dans le tableau **t**.

Voici l'en-tête de la méthode :

```
public static int getNombreValeursPresentes(int[] t, int[] v)
```

Exercice 3 :

Proposez une méthode qui reçoit un tableau d'entiers **t**, un nombre entier **x** et une position et qui insère le nombre **x** dans le tableau **t** à la position spécifiée en décalant les valeurs d'une case vers la fin du tableau (la dernière valeur du tableau sera perdue).

Voici l'en-tête de la méthode:

```
public static int inserer(int[] t, int x, int position)
```

Exercice 4 :

Proposez une méthode qui reçoit un tableau d'entiers **t** trié par ordre croissant et un nombre entier **x** et qui insère le nombre **x** dans le tableau **t** à la bonne position pour que le tableau reste trié (la dernière valeur du tableau sera perdue).

Voici l'en-tête de la méthode :

```
public static int inserer(int[] t, int x)
```

Laboratoire Préparation à l'intra #1

Exercice 5 :

La méthode suivante effectue une recherche binaire dans un tableau de chaînes de caractères. La méthode retourne -1 si la valeur cherchée n'est pas dans le tableau et retourne l'indice de la valeur si elle est trouvée.

On vous demande de compléter les trous par le code approprié.

```
public static int rechercheBinaire(String[] tableau, String valeurCherchee) {

    boolean trouvee = false;
    int debut = 0;
    int fin = _____; //dernier indice
    int milieu = -1;

    /*Tant que l'espace de recherche contient au moins une case et qu'on n'a pas
    encore trouvé la valeur cherchée : */
    while (_____ && !trouvee) {
        milieu = _____; //milieu de l'espace de recherche
        if (_____) { //les chaines sont égales
            trouvee = true;
        }
        else if (tableau[milieu].compareTo(valeurCherchee) > 0) {
            //on conserve la 1re moitié de l'espace de recherche :
            fin = _____;
        }
        else {
            //on conserve la 2e moitié de l'espace de recherche :
            debut = _____;
        }
    }

    //On retourne milieu ou -1, selon qu'on a trouvé la valeur cherchée ou non :
    if (trouvee)
        return _____;
    else
        return _____;
}
```