

## Langage Java, exercices série 4

---

### Recherche linéaire dans un tableau non trié

Objectif : Ecrire un programme Java effectuant une recherche séquentielle dans un tableau linéaire (une dimension) non trié.

Spécifications de l'algorithme :

- Soit  $t$  un tableau d'entiers de  $n$  éléments non rangés (de 0 à  $n-1$ ).
  - On recherche le rang (la place) de l'élément **elt** dans ce tableau. L'algorithme renvoie le rang (entre 1 et  $n$ ).
- (La valeur **-1** est renvoyée lorsque l'élément **elt** n'est pas présent dans le tableau  $t$ ).

Version **Tantque** avec "et alors" (opérateur **et** optimisé)

```
i ← 0 ;
Tantque (i < n) et alors (t[i] ≠ elt) faire
    i ← i+1
finTant;

si i < n alors
    rang ← i+1
sinon
    rang ← -1
Fsi
```

Version **Tantque** avec "et" (opérateur **et** non optimisé)

```
i ← 0 ;
Tantque (i < n-1) et (t[i] ≠ elt) faire
    i ← i+1
finTant;

si t[i] = elt alors
    rang ← i+1
sinon
    rang ← -1
Fsi
```

Version **Pour** avec instruction de sortie (Sortirsi)

```

pour i ← 0 jusqu'à n-1 faire
    Sortirsi t[i] = elt
fpour;

si i ≤ n alors
    rang ← i+1
sinon
    rang ← -1
Fsi

```

Traduire chacune des trois versions sous forme d'une méthode Java.

Proposition de squelette de classe Java à implanter :

```

public class ApplicationRechLin
{
    static int max = 20;
    static int[] table = new int[max] ; // 20 cellules à examiner de 0 à 20

    static void AfficherTable (int[] t )
    {
        // Affichage du tableau
        int n = t.length;
        for ( int i = 0; i < n; i++)
            System.out.print(t[i]+" , ");
        System.out.println();
    }

    static void InitTable ( )
    {
        // remplissage aléatoire du tableau
        int n = table.length;
        for ( int i = 0; i < n; i++)
        {
            table[i] = (int)(Math.random()*100);
        }
    }

    static int RechSeq1( int[] t, int elt )
    {
    }

    static int RechSeq2( int[] t, int elt )
    {
    }

    static int RechSeq3( int[] t, int elt )
    {
    }

    public static void main(String[] args)
    {
        InitTable ( );
        System.out.println("Tableau initial :");
        AfficherTable (table );
        int rang1, rang2, rang3;
        int x = SimpleDataInput.readInt();
    }
}

```

```
//appeler ici les méthodes de recherche ...

if (rang1 > 0)
{
    System.out.println("Elément " + x + " trouvé en : " + rang1);
    System.out.println("Elément " + x + " trouvé en : " + rang2);
    System.out.println("Elément " + x + " trouvé en : " + rang3);
}
else
    System.out.println("Elément " + x + " non trouvé !");
}
```