

**DSTI2  
AP4  
TP N° 3****Exercice 1**

Implémentez ces différentes fonctions

- 1) Cette fonction permet de remplir les éléments d'un tableau **t**.

```
static void saisir(int[] t);
```

- 2) Cette fonction renvoie la somme des éléments du tableau **t**

```
static int somme(int[] t);
```

- 3) Cette fonction renvoie le maximum des éléments du tableau **t**

```
static int maximum(int[] t);
```

- 4) Cette fonction renvoie la différence entre le maximum et le minimum des éléments du tableau **t**.

```
static int delta(int[] t);
```

- 5) Cette fonction fait le tri des éléments du tableau **t**.

```
static void bubble(int[] t);
```

- 6) Cette fonction renvoie vrai si les deux tableaux contiennent exactement les mêmes éléments aux mêmes indices.

```
static boolean egalite(int[] t1, int[] t2);
```

- 7) Cette fonction renvoie vrai si les deux tableaux contiennent les mêmes éléments ; les deux tableaux ne sont pas forcément de même taille.

```
static boolean equivalence(int[] t1, int[] t2);
```

- 8) Faites le test de toutes ces fonctions dans une fonction main.

## Exercice 2

On appelle palindrome un mot pouvant se lire indifféremment dans les deux sens : par exemple *Laval*, *ici*, le mot vide, etc. Plus formellement, un mot de longueur  $L$  est un palindrome si pour tout  $i < L$ , le caractère à la position  $i$  est égal au caractère à la position  $L - i - 1$ .

Cette notion peut également s'appliquer à une séquence d'entiers.

1. Écrivez sur papier un algorithme vérifiant si un mot est un palindrome.
2. Écrivez une fonction en Java prenant en argument un tableau d'entier et renvoyant un booléen indiquant si le tableau est un palindrome. Vous écrirez également la méthode statique `main` permettant de tester cette fonction.
3. En utilisant la méthode `charAt`, écrivez une autre fonction prenant cette fois ci une **String** en argument et vérifiant si c'est un palindrome. La chaîne vérifiée sera récupérée par la méthode `main` sur la ligne de commande.

## Exercice 3

Il s'agit d'un jeu qui consiste à choisir un nombre au hasard et le faire deviner à l'adversaire en lui disant « *plus grand* » ou « *plus petit* ». Ecrire un programme qui devine un nombre entre 1 et 100 choisi par l'utilisateur.

Le programme utilisera au moins 2 variables de type `int` : **inf** et **sup** initialisées respectivement à 0 et 100.

Elles seront mises à jour en maintenant le fait que la valeur à deviner sera toujours comprise entre **inf** et **sup**.

### Exemple d'exécution

**java Devin**

**Est-ce 50 ? plus**

**Est-ce 75 ? plus**

**Est-ce 87 ? moins**

**Est-ce 81 ? plus**

**Est-ce 84 ? moins**

**Est-ce 82 ? oui**

Utiliser la classe `Scanner` pour la saisie des valeurs.