TD 0

Tableaux

Exercice 1 : Initialisation de tableaux

- (1) Donner un exemple de code qui crée un tableau et l'initialise. Prendre soin d'expliquer chaque étape en dessinant l'état de la mémoire.
- (2) Faire de même avec un tableau à deux dimensions.

Exercice 2 : Analyse d'un programme sur les tableaux

Analyser et dessiner l'état de la mémoire lors de l'exécution du code suivant:

```
int a[] = new int[2];
int b[] = new int[2];
a[0] = 0;
a[1] = 1;
System.out.println( "a : " + a[0] + " , " + a[1] );
System.out.println( "b : " + b[0] + " , " + b[1] );
b[0] = 2;
b[1] = 3;
b = a;
b[0] = 100;
System.out.println( "a : " + a[0] + " , " + a[1] );
System.out.println( "b : " + b[0] + " , " + b[1] );
```

Exercice 3 : Fonctions classiques sur les tableaux

- (1) Écrire une fonction `affiche(int[] t)` qui affiche les éléments du tableau `t`.
- (2) Écrire une fonction `renverser(int[] t)` qui renverse l'ordre des éléments du tableau `t`, et ce, sans créer de tableau auxiliaire.
- (3) Écrire une fonction `int min_tableau(int[] t)` qui renvoie le plus petit élément du tableau `t`.
- (4) Écrire une fonction `int moyenne_tableau(int[] t)` qui renvoie la moyenne des entiers contenus dans le tableau `t`.
- (5) Écrire une fonction `int[] occurrences(int[] t)` qui renvoie un nouveau tableau `s` tel que `s[i]` est égal au nombre d'occurrences de `i` dans le tableau `t`.

Exercice 4 : Le jeu du sudoku

Une grille de sudoku est un tableau d'entiers à neuf lignes et neuf colonnes. Chaque élément est compris entre 1 et 9. Pour être correcte, la grille doit vérifier les contraintes suivantes :

- Chaque ligne doit contenir tous les entiers de 1 à 9,
- Chaque colonne doit contenir tous les entiers de 1 à 9,
- Tous les carrés de hauteur et largeur 3, dont les sommets Nord-Ouest sont situés aux coordonnées $(0,0)$, $(0,3)$, $(0,6)$, $(3,0)$, $(3,3)$, $(3,6)$, $(6,0)$, $(6,3)$, $(6,6)$ doivent contenir tous les entiers de 1 à 9.

Écrire un programme qui permet de vérifier si une grille de sudoku est correctement remplie.

Exercice 5 : Le jeu du morpion

Coder le jeu du morpion.

Prendre soin d'analyser les structures de données à utiliser avant de programmer. Écrire le code le plus lisible possible en factorisant le code à l'aide de fonctions. Il est en effet nécessaire de séparer les phases d'affichage, de récupération de donnée et de calcul.