

**J1MI2013: Algorithmes et Programmes: feuille 7****Tableaux(4)****Travaux dirigés**

**Exercice 1.** On considère un tableau d'entiers et on souhaite écrire une fonction `supprimerOccurrences` permettant d'enlever de la suite toutes les occurrences d'un élément  $x$  passé en paramètre. La fonction devra de plus renvoyer le nombre d'occurrences de  $x$  supprimées.

Exemple : si le tableau contient la suite :

$$2, -7, 4, 5, 12, 10, 4, 2, 4, 132, -18, 19$$

et si  $x = 4$ , la fonction retournera 3 et après son exécution le tableau contiendra la suite :

$$2, -7, 5, 12, 10, 2, 132, -18, 19$$

On envisagera deux versions de la fonction :

- une première qui appelle la fonction `supprimerPremiereOccurrence` autant de fois que nécessaire
- et une deuxième qui n'effectue qu'un seul parcours du tableau.

Comparer le nombre de décalages nécessaires dans les deux versions (on étudiera aussi le cas où tous les éléments du tableau sont égaux à  $x$ ).

**Exercice 2.** Soient `t1` et `t2` deux tableaux de nombres entiers triés dans l'ordre croissant. On veut fusionner ces deux tableaux en un troisième tableau `t3` également trié dans l'ordre croissant. Par exemple, si `t1=[2, 5, 5, 8, 9]` et `t2=[1, 3, 5]`, le tableau résultant `t3` contiendra `[1, 2, 3, 5, 5, 5, 8, 9]`.

Écrire la fonction `int fusion(int t1[], int n1, int t2[], int n2, int t3[])` qui fusionne les `n1` éléments (triés) de `t1` et les `n2` éléments (triés) de `t2` dans le tableau `t3`. On suppose que la taille de `t3` est suffisante. La fonction retournera la taille du tableau `t3`.

Étudier le nombre de comparaisons et d'affectations entre éléments de tableaux effectuées par votre algorithme.