

## 4TPM209U: Algorithmique des tableaux

### TP Noté

## Travaux dirigés sur machine

*Rappel : le fichier `bibTableau.py` doit se trouver dans le même répertoire que vos programmes Python.*

### Exercice 1 – Recherche dans un tableau

Écrire une fonction Python `trouverIndicesElement(t, n, elt, indices)` qui trouve les indices de toutes les occurrences de `elt` dans le tableau `t`, les stocke dans le tableau `indices` et retourne `nI` le nombre de valeurs dans le tableau `indices`.

Par exemple, avec `t = [5, 7, 12, 5, 5, 2, 6]` et `elt = 5`, `trouverIndicesElement(t, 7, elt, indices)` retourne 3 et le tableau `indices` contient alors les valeurs `[0, 3, 4]`.

### Exercice 2 – Manipulation de tableau

Écrire une fonction `supprimerPremiereOccurrence(t, n, elt)` permettant d'enlever du tableau `t` la première occurrence d'un élément `elt` passé en paramètre (le tableau `t` ne sera pas modifié si `elt` n'appartient pas à `t`). L'ordre initial des éléments du tableau sera conservé. La fonction doit renvoyer le nouveau nombre d'éléments dans le tableau.

### Exercice 3 – Récursivité

Écrire une fonction récursive `nbChiffresK(n, k)` qui prend en entrée un entier `n` strictement positif et un chiffre `k` et retourne le nombre de chiffres `k` présent dans `n`. Par exemple l'appel `nbChiffresK(255847952, 5)` retournera 3 car il y a 3 fois le chiffre 5 dans 255847952 et `nbChiffresK(255847952, 1)` retournera 0 car il n'y a pas le chiffre 0 dans 255847952.

**Exercice 4** – Une suite de nombres est *palindrome* si on peut la lire indifféremment de gauche à droite ou de droite à gauche. Soit `t` un tableau de `n` nombres, on veut tester s'il constitue une suite palindrome.

1. Écrire une fonction récursive `estPalindromeSectionRec(t, i, j)` qui renvoie `True` si la section du tableau `t` entre les indices `i` et `j` inclus constitue une suite palindrome et renvoie `False` sinon. On supposera, par convention, que la fonction renvoie `True` si `i > j`.

Exemples : Si `t = [3, 7, 2, 1, 1, 2, 5]` l'appel `estPalindromeSectionRec(t, 2, 5)` doit renvoyer `True`, alors que l'appel `estPalindromeSectionRec(t, 0, 6)` doit renvoyer `False`.

2. Écrire une fonction `estPalindromeTab(t, n)` qui fait appel à la fonction précédente pour tester si les `n` nombres contenus dans `t` constituent une suite palindrome.