

---

**TD 3 (algorithmique) - Matrices et chaînes.**


---

MATRICES

Dans les exercices qui suivent, on suppose que toutes les lignes d'une matrice contiennent le même nombre d'éléments. On suppose également, pour simplifier, que les matrices examinées ne sont pas vides.

**Exercice 1**

Une matrice est *carrée* si elle contient autant de lignes que de colonnes. La *trace* d'une matrice carrée  $M$  est la somme des éléments de sa diagonale principale, c'est-à-dire  $\sum_{i=0}^{n-1} M[i][i]$ . Écrivez une fonction renvoyant la trace d'une matrice carrée donnée.

**Exercice 2**

Une matrice est *symétrique* si elle est carrée et si  $M[i][j] = M[j][i]$  pour toute paire d'indices  $i$  et  $j$ . Écrivez une fonction vérifiant si une matrice  $M$  donnée est symétrique.

**Exercice 3**

La *transposée* d'une matrice  $M$  est la matrice  $M^t$  obtenue en échangeant les lignes et les colonnes de  $M$ . Par exemple :

$$\text{si } M = \begin{array}{|c|c|c|c|} \hline 3 & 1 & 4 & 2 \\ \hline 2 & 5 & 6 & 2 \\ \hline 3 & 6 & 7 & 1 \\ \hline \end{array}, \text{ alors } M^t = \begin{array}{|c|c|c|} \hline 3 & 2 & 3 \\ \hline 1 & 5 & 6 \\ \hline 4 & 6 & 7 \\ \hline 2 & 2 & 1 \\ \hline \end{array}$$

Écrivez une fonction qui renvoie la transposée d'une matrice  $M$  donnée.

**Exercice 4**

Écrivez une fonction qui transpose une matrice carrée donnée *en place*, c'est-à-dire que la fonction ne renvoie rien et qu'après son appel, la matrice passée en paramètre est transposée.

**Exercice 5**

Adaptez votre réponse à l'exercice 4 pour pouvoir transposer la matrice en place même quand elle n'est pas carrée.

CHAÎNES

**Exercice 6**

Un mot est un *anagramme* d'un autre mot s'il contient exactement les mêmes lettres mais dans un ordre différent. Par exemple, '**imaginer**' est un anagramme de '**migraine**'. Écrivez une fonction prenant deux chaînes  $s$  et  $t$  en paramètres et renvoyant **True** si l'une est un anagramme de l'autre, et **False** sinon.

Rappel : la fonction `ord(c)`, qui peut vous être utile, renvoie la valeur numérique du caractère  $c$ .

**Exercice 7**

Le *run length encoding* (RLE) est une méthode de compression de données qui réduit la taille d'un texte en résumant les occurrences successives de caractères par une paire  $(a, n_a)$ , où  $a$  est un caractère dans le texte et  $n_a$  le nombre d'occurrences consécutives de  $a$  dans ce texte à partir d'une position donnée. Par exemple, l'algorithme RLE appliqué à la chaîne '**aaaaabbbbcbcbddddddeaeae**' donnera le texte '**a6b4c1b2c1b2d5e1a2e3a1e1**'.

1. Écrivez une fonction renvoyant l'encodage RLE d'une chaîne de caractères donnée (on suppose que cette chaîne ne contient pas de chiffres).
2. Rajouter un 1 quand un caractère n'apparaît qu'une seule fois augmente la taille du texte. Modifiez votre fonction pour qu'elle n'enregistre pas les 1 dans le résultat (dans le cas de l'exemple précédent, on renverrait cette fois '**a6b4cb2cb2d5ea2e3ae**').