
Dans cet exercice, étant donné un tableau de tableaux d'entiers `tab` (qu'on rappelle être de type `int[][]`), on appellera *lignes* les tableaux `tab[0]`, `tab[1]`, etc. Ainsi par exemple, la ligne 0 est composée des éléments `tab[0][0]`, `tab[0][1]`, etc.

On définit de façon similaire les *colonnes* de `tab`. Par exemple la colonne 0 est composée des éléments `tab[0][0]`, `tab[1][0]`, etc.. Attention, les lignes de `tab` n'ont pas nécessairement toutes la même longueur.

Exercice 1 : Manipulations de base


1. Écrire une fonction `afficheTab` qui prend en entrée un tableau de tableaux d'entiers et qui l'affiche ligne par ligne.
2. Écrire une fonction `compareTab` qui prend en entrée deux tableaux de tableaux d'entiers et qui renvoie un booléen indiquant si ces tableaux sont identiques.
3. Écrire une fonction `copieTab` qui prend en entrée un tableau de tableaux d'entiers et qui en renvoie une copie.

Exercice 2 : Carré magique

1. Écrire une fonction `sommeLigne` qui prend en entrée un tableau de tableaux d'entiers `tab` ainsi qu'un entier `i` et qui renvoie la somme des éléments de la ligne `i`. Par convention, si la ligne `i` est vide alors la fonction renverra 0.
2. Écrire une fonction `sommeColonne` qui prend en entrée un tableau de tableaux d'entiers `tab` ainsi qu'un entier `i` et qui renvoie la somme des éléments de la colonne `i`. On fera attention à ne considérer que les lignes qui possèdent une case `i`.
3. Écrire une fonction `estMagiqueLigne` qui prend en entrée un tableau de tableaux d'entiers `tab` et qui renvoie un booléen indiquant si toutes les lignes de `tab` ont la même somme.
4. Écrire une fonction `estMagiqueColonne` qui prend en entrée un tableau de tableaux d'entiers `tab` et qui renvoie un booléen indiquant si toutes les colonnes de `tab` ont la même somme. On pourra commencer par calculer le nombre de colonnes, qui est égal à la longueur de la plus grande ligne.
5. Écrire une fonction `estCarre` qui prend en entrée un tableau de tableaux d'entiers `tab` et qui renvoie un booléen indiquant si toutes les lignes de `tab` ont la même longueur que le tableau `tab` lui-même.
6. Écrire une fonction `estMagique` qui prend en entrée un tableau de tableaux d'entiers `tab` et qui renvoie un booléen indiquant si `tab` est magique, c'est à dire si `tab` est carré et que les sommes de chacune des lignes, des colonnes et des deux diagonales sont identiques.

Exercice 3 : Défis

1. (**challenge**) Écrire une fonction `estLatin` qui prend en entrée un tableau de tableaux d'entiers `tab` et qui renvoie un booléen indiquant si `tab` est un carré et si sur chaque ligne et chaque colonne les entiers de 1 à n (où n est la taille de `tab`) apparaissent exactement une fois chacun.

2. (**challenge**) Écrire une fonction `estUneSequence` qui prend en entrée un tableau de tableaux d'entiers `tab` et qui renvoie un booléen indiquant si `tab` est un carré qui contient exactement une fois chaque entier compris entre 1 et n^2 inclus (où n est la taille de `tab`).
3. () Écrire une fonction `estSudoku` qui prend en entrée un tableau de tableaux d'entiers `tab` et un entier `n`, et qui renvoie un booléen.
`estSudoku(tab,n)` doit renvoyer `true` si et seulement si
- `tab` est un carré de côté n^2
 - `tab` est un sudoku généralisé : c'est un carré latin (cf. question 1), et si l'on divise `tab` en n^2 blocs de taille `n` sur `n`, chacun de ces blocs contient exactement une fois chaque entier compris entre 1 et n^2 .

On pourra commencer par résoudre le cas familier $n = 3$.