

Le but de ce TP est de s'intéresser aux rectangles magiques que l'on représentera par des tableaux à deux dimensions.

Exercice 1 : Affichage d'un tableau à deux dimensions

Écrire une fonction `afficheTab2D` qui prend en paramètre un tableau d'entiers à deux dimensions de taille arbitraire `tab` et qui l'affiche en renvoyant à la ligne entre chaque ligne du tableau.

Exercice 2 : Vérification si un rectangle est magique

Un rectangle est *magique* si

- les sommes des éléments de chaque ligne sont égales,
- et les sommes des éléments de chaque colonne sont égales.

FIGURE 1 – Exemple de rectangle magique de taille 3×7

1	12	20	8	17	6	13
14	2	10	21	9	16	5
18	19	3	4	7	11	15

2.a] Écrire une fonction `verifierLignes` qui prend en paramètre un tableau d'entiers à deux dimensions de taille arbitraire `tab` et qui renvoie un booléen indiquant si les sommes des éléments de chaque ligne de `tab` sont égales (Indication : on peut utiliser une fonction annexe `sommeLigne` qui prend en paramètres `tab` et un indice de ligne et qui renvoie la somme des éléments de la *i*-ème ligne de `tab`).

Par exemple, pour `tab` comme indiqué dans la Fig. 1, la fonction renverra `true`.

2.b] Sur le même principe, écrire une fonction `verifierColonnes`.

Par exemple, pour `tab` comme indiqué dans la Fig. 1, la fonction renverra `true`.

2.c] Écrire une fonction `estMagique` qui prend en paramètre un tableau d'entiers à deux dimensions de taille arbitraire `tab` et qui renvoie un booléen indiquant si `tab` est un rectangle magique.

Par exemple, pour `tab` comme indiqué dans la Fig. 1, la fonction renverra `true`.

Exercice 3 : Vérifier si un rectangle est normal

Un rectangle est *normal* si les nombres de la grille vont de 1 à mn , autrement dit si tous les nombres compris entre 1 et mn n'apparaissent qu'une seule fois dans le rectangle. Écrire une fonction `estNormal` qui prend en paramètre un tableau d'entiers à deux dimensions de taille arbitraire `tab` et qui renvoie un booléen indiquant si `tab` est un rectangle normal.

Par exemple, pour `tab` comme indiqué dans la Fig. 1, la fonction renverra `true`.

Exercice 4 : Transformations de rectangle magique

4.a] Écrire une fonction `sommeRectangles` qui prend en paramètres deux tableaux d'entiers `tab1` et `tab2` à deux dimensions et de même taille et qui renvoie un nouveau tableau de même taille qui contient la somme case par case des deux tableaux passés en paramètres.

4.b] Écrire une fonction `symetrieHorizontale` qui prend en paramètre un tableau d'entiers à deux dimensions de taille arbitraire `tab` et qui renvoie un nouveau tableau de même taille qui est la symétrie horizontale du tableau `tab`. Exemple sur le rectangle magique de la Fig. 1 :

18	19	3	4	7	11	15
14	2	10	21	9	16	5
1	12	20	8	17	6	13

4.c] (Challenge) Écrire une fonction `symetrieVerticale` qui prend en paramètre un tableau d'entiers à deux dimensions de taille arbitraire `tab` et qui renvoie un nouveau tableau de même taille qui est la symétrie verticale du tableau `tab`. Exemple sur le rectangle magique de la Fig. 1 :

13	6	17	8	20	12	1
5	16	9	21	10	2	14
15	11	7	4	3	19	18

4.d] (Challenge) Écrire une fonction `transpose` qui prend en paramètre un tableau d'entiers à deux dimensions de taille arbitraire `tab` et qui renvoie un nouveau tableau de taille inversée pour lequel lignes et colonnes sont inversées. Exemple sur le rectangle magique de la Fig. 1 :

1	14	18
12	2	19
20	10	3
8	21	4
17	9	7
6	16	11
13	5	15

4.e] (Challenge) Écrire une fonction `rotation` qui prend en paramètre un tableau d'entiers à deux dimensions de taille arbitraire `tab` et qui renvoie un nouveau tableau qui est la rotation d'un quart de tour dans le sens contraire des aiguilles d'une montre (sens trigonométrique!). Exemple sur le rectangle magique de la Fig. 1 :

13	5	15
6	16	11
17	9	7
8	21	4
20	10	3
12	2	19
1	14	18