Exercice 1. — Écrire une fonction `occurences(v, tab)` qui renvoie le nombre d'occurences de la valeur `v` dans le tableau `tab`.

Exercice 2. — Écrire une fonction `echange(tab, i, j)` qui échange dans le tableau `tab` les éléments aux indices `i` et `j`.

Exercice 3. — Écrire une fonction `somme(tab)` qui renvoie la somme des éléments d'un tableau d'entiers.
En déduire une fonction `moyenne(tab)` qui calcule et renvoie la moyenne des éléments d'un tableau `tab` supposé non vide, 0 sinon.

Exercice 4. — Écrire une fonction `produit(tab)` qui calcule et renvoie le produit des éléments d'un tableau d'entiers `tab` supposé non vide. Si le tableau contient 0, la fonction devra renvoyer 0 sans terminer le calcul.

Exercice 5. — Écrire une fonction `mirroir(tab)` qui reçoit un tableau en paramètre et le modifie pour échanger le premier élément avec le dernier, le second avec l'avant-dernier, etc. Dit autrement, on remplace le tableau par son image miroir. On pourra se servir de la fonction `echange` précédemment codée.

Exercice 6. — Écrire une fonction `tab_entiers` qui génère et renvoie un tableau de 100 entiers tirés au hasard entre 1 et 1000.
On utilisera la fonction `randrange(a, b)` de la librairie `random`. Pour pouvoir utiliser cette fonction, il faut préalablement importer la librairie `random` à l'aide de l'instruction : `from random import randrange`.

Exercice 7. — Écrire une fonction `tab_aleatoire(n, a, b)` qui renvoie un tableau de taille `n` contenant des entiers tirés au hasard entre `a` et `b`.

Exercice 8. — Pour mélanger les éléments d'un tableau aléatoirement, il existe un algorithme très simple (algorithme de randomisation directe) qui procède ainsi : on parcourt le tableau de la gauche vers la droite et, pour chaque élément à l'indice `i`, on l'échange avec un élément situé à un indice tiré aléatoirement entre 0 et `i` (inclus).
Écrire une fonction `randomisation_directe(tab)` qui réalise cet algorithme.
On pourra se servir de la fonction `echange` codée précédemment. Cet algorithme s'appelle le mélange de Knuth.

Exercice 9. — Écrire une fonction `suffixe(tab1, tab2)` qui renvoie `True` si le tableau `tab1` est un suffixe du tableau `tab2`, c'est-à dire si le tableau `tab2` termine par les éléments du tableau `tab1` dans le même ordre.

Exercice 10. — Écrire une fonction `hamming(tab1, tab2)` qui prend en paramètres deux tableaux, que l'on supposera de même taille, et qui renvoie le nombre d'indices auxquels les deux tableaux diffèrent.