

Dans l'ensemble de ce TP, sauf mention contraire, le terme « liste » désignera une **liste simplement chaînée**.

**Exercice 1.** — Écrire une fonction `listeN(n)` qui reçoit en paramètre un entier  $n$ , supposé positif ou nul, et renvoie la liste des entiers  $1, 2, \dots, n$  dans cet ordre.

Si  $n = 0$ , la liste renvoyée est vide.

**Exercice 2.** — Écrire une fonction `affiche_liste(lst)` qui affiche, en utilisant la « fonction » `print`, tous les éléments de la liste `lst`, séparés par le motif « -> ».

**Exercice 3.** — Écrire une fonction `occurrences(x, lst)` qui renvoie le nombre d'occurrences de la valeur  $x$  dans la liste `lst`.

**Exercice 4.** — Écrire une fonction `trouve(x, lst)` qui renvoie le rang de la première occurrence de  $x$  dans `lst` le cas échéant, `None` sinon.

**Exercice 5.** — Écrire une fonction `identiques(l1, l2)` qui renvoie un booléen indiquant si les listes `l1` et `l2` sont identiques, c'est-à-dire si elles contiennent exactement les mêmes éléments, dans le même ordre.

On suppose que l'on peut comparer les éléments de `l1` et `l2` avec l'opérateur `==` de PYTHON.

**Exercice 6.** — Écrire une fonction `insérer(x, lst)` qui prend en paramètres un entier  $x$  et une liste d'entiers `lst`, supposée triée par ordre croissant, et qui renvoie une nouvelle liste dans laquelle  $x$  a été inséré à sa place. Ainsi, insérer la valeur 3 dans la liste `1,2,5,8` renvoie la liste `1,2,3,5,8`.

**Exercice 7.** — Écrire une fonction `liste_de_tableau(t)` qui renvoie une liste qui contient les éléments du tableau `t`, dans le même ordre.

**Exercice 8.** — Écrire une fonction `derniere_cellule(lst)` qui renvoie la dernière cellule de la liste `lst`. On suppose la liste `lst` non vide.