

## TD 3 (programmation) - Listes et boucles `for`.

Comme d'habitude, les exercices suivants doivent être réalisés sans utiliser d'éventuelles fonctions toutes faites qui répondraient directement aux questions.

### Exercice 1 (Liste de 1 à 100)

On cherche à fabriquer la liste des nombres de 1 à 100 de trois façons possibles. Lesquelles fonctionnent? Corrigez celles qui sont fausses.

```
1 lst1 = []
2 for i in range(100):
3     lst1[i] = i + 1
4 lst2 = []
5 for i in range(100):
6     lst2.append(i+1)
7 lst3 = list(range(100))
```

### Exercice 2 (Liste des nombres de 1 à $n$ )

Répondez aux deux sous-questions suivantes sans transtyper `range`.

- Écrivez un programme qui demande un naturel  $n$  à l'utilisateur, crée la liste des nombres de 1 à  $n$  inclus, puis l'affiche.
- Même question mais avec seulement les nombres pairs entre 1 et  $n$ .

### Exercice 3 (Tous pairs?)

Dans cet exercice, on travaille sur une liste d'entiers  $L$ , que l'on suppose déjà initialisée.

- Écrivez un code qui parcourt la liste et affiche chacun de ses éléments un par un suivi de `Pair` s'il est pair et `Impair` sinon.
- Écrivez un code qui affiche `Tous pairs` si tous les entiers de  $L$  sont pairs, et `Pas tous pairs` s'il y a au moins un nombre impair dedans.

#### Indices

Attention à ne pas confondre  $\exists$  et  $\forall$ : s'il existe un nombre impair dans la liste, on peut directement conclure que ses éléments ne sont pas tous pairs. Mais pour pouvoir dire que tous les éléments sont pairs, il faut les avoir tous lus. Le plus simple est de supposer que tous les éléments sont pairs, puis de chercher une contradiction (un élément impair) dans la liste.

### Exercice 4 (Énumération conditionnelle)

Écrivez un code qui affiche le nombre d'entiers supérieurs ou égaux à 10 dans une liste d'entiers donnée.

#### Indices

Utilisez un compteur initialement nul, incrémenté à la rencontre de chaque élément  $\geq 10$ .

### Exercice 5 (Somme d'une liste)

Écrivez un code qui affiche la somme de tous les nombres d'une liste  $L$ . Par exemple, si  $L = [1, 2, 3]$ , le programme affichera 6.

**Exercice 6 (Renversement de liste)**

- (a) Écrivez un code qui renverse la liste  $L$  (pré-initialisée) dans la liste  $R$ . Par exemple, si  $L = [1, 2, 3]$ , après avoir exécuté votre code,  $R$  vaudra  $[3, 2, 1]$ .

**Indices**

Rajouter chaque élément de  $L$  lue à l'envers à la fin de  $R$ , initialement vide.

- (b) Même question, mais on demande cette fois-ci de renverser  $L$  **en place**, c'est-à-dire sans utiliser de liste auxiliaire.

**Indices**

Échanger le premier et le dernier élément de  $L$  nous met sur la piste du résultat à obtenir.

**Exercice 7 (Inclusion)**

Écrivez un code qui teste si tous les nombres de la liste  $L$  appartiennent à la liste  $R$  (toutes deux déjà initialisées), et affiche le résultat du test.