CORRIGÉ

1re

Algorithmique — listes en Python

Corrigé détaillé
6 exercices

> Ce qui est noté au contrôle, ce n'est pas le résultat seul : c'est le raisonnement posé, puis le calcul mené jusqu'au bout avec l'unité.

1 Indices

1. `L[1]` vaut **8** (l'indice commence à 0).
2. `L[-1]` vaut **9** (dernier élément).
3. `len(L)` vaut **4**.

2 Compréhension

1. k prend les valeurs 0, 1, 2, 3.
2. La liste est `[1, 3, 5, 7]` : les quatre premiers nombres impairs.

3 Moyenne

1. `def moyenne(L):` / `s = 0` / `for x in L:` / `s = s + x` / `return s / len(L)`
2. Test : `moyenne([10, 12, 17])` renvoie `13.0`.

4 Maximum

1. `def maximum(L):` / `m = L[0]` / `for x in L:` / `if x > m:` / `m = x` / `return m`
2. On initialise avec le premier élément, pas avec 0 (la liste peut ne contenir que des négatifs).

5 Simulation

1. La liste contient **100 lancers simulés** d'un dé équilibré.
2. Pour compter les 6 : `n = 0`, puis `for x in L:` / `if x == 6: n = n + 1`.
3. La fréquence `n / 100` approche $\frac{1}{6}$ lorsque le nombre de lancers augmente.

6 Suite et seuil

1. `def termes_seuil():` / `L = [1]` / `while L[-1] * 1.5 < 100:` / `L.append(L[-1] * 1.5)` / `return L`
2. Termes : 1 ; 1,5 ; 2,25 ; 3,375 ; 5,0625 ; 7,59… ; 11,39… ; 17,09… ; 25,63… ; 38,44… ; 57,67… ; 86,50…
3. Le terme suivant, 129,7…, dépasse 100 : la liste compte **12 termes**.

Corrige-toi toi-même. Compte tes points exercice par exercice : c'est ainsi qu'on voit où l'on perd, et ce n'est presque jamais là où on le croit.

Ce qui est évalué	Points	Ce qui fait perdre le point
Construire une liste	2 pts	Décalage d'indice ou de range
Écrire une boucle de parcours	2 pts	Mauvaise initialisation
Tester le programme	1 pt	Résultat non vérifié à la main