

# TP noté MPSI 1

Informatique pour tous, première année

Julien REICHERT

Exercice 1 : Résoudre sur Codewars l'exercice à la page ci-dessous.

<https://www.codewars.com/kata/5a3dd29055519e23ec000074>

Exercice 2 : Résoudre sur Codewars l'exercice à la page ci-dessous.

<https://www.codewars.com/kata/5808dcb8f0ed42ae34000031/>

Exercice 3 : Résoudre sur Codewars l'exercice à la page ci-dessous.

<https://www.codewars.com/kata/58f8a3a27a5c28d92e000144/>

Exercice 4 : Écrire en Python une fonction `debordement(l, c)` qui prend en entrée une liste d'entiers `l` et un entier positif `c` et qui détermine si toutes les sommes partielles des éléments de `l` entre le début et n'importe quel indice restent dans l'intervalle entre 0 et `c` inclus.

Par exemple, l'appel `debordement([1, -3, 4], 5)` retournera `False` car la somme des deux premiers éléments est négative, mais l'appel `debordement([4, -3, 4], 5)` retournera `True` car les sommes partielles sont toujours positives et inférieures ou égales à 5.

Exercice 5 : Écrire en Python une fonction qui calcule la factorielle de l'entier en argument, et en déduire une fonction qui calcule le coefficient binomial de paramètres les deux entiers en argument.

Pour préciser, la fonction `kparmin(5, 2)` retourne 10, et on demande que l'ordre des arguments donnant `k` parmi `n` soit le suivant : `kparmin(n, k)`.

Exercice 6 : Résoudre sur Codewars l'exercice à la page ci-dessous.

<https://www.codewars.com/kata/5899dc03bc95b1bf1b0000ad>

Exercice 7 : Résoudre sur Codewars l'exercice à la page ci-dessous.

<https://www.codewars.com/kata/57e921d8b36340f1fd000059>

Exercice 8 : Résoudre sur Codewars l'exercice à la page ci-dessous.

<https://www.codewars.com/kata/55eea63119278d571d00006a>

Exercice 9 : Écrire en Python une fonction `correspondances(l, ll)` qui détermine le nombre de fois où un élément de la liste ou de la chaîne de caractères `l` est égal à l'élément de même indice dans la liste ou la chaîne de caractères `ll`.

Par exemple, l'appel `correspondances("chercher", "cher")` retourne 4 car les quatre premiers caractères des deux chaînes sont communs, mais l'appel `correspondances([1, 4], [1, 2, 3, 1, 4])` retourne 1 car ce ne sont que les deux premiers éléments de chaque liste qui sont comparés.

Exercice 10 : Écrire en Python une fonction `correspondancesmax(1, 11)`, variante de la fonction de l'exercice précédent, qui détermine le nombre maximal de fois où un élément de `l` est égal à un élément de `ll` en commençant à comparer avec le décalage qui fournirait la plus grande valeur possible pour la fonction.

Dans ce cas, l'appel `correspondancesmax([1, 4], [1, 2, 3, 1, 4])` retourne bien 2, et par ailleurs l'appel `correspondancesmax([1, 2, 3, 4, 5], [2, 3, 5])` retourne 2, vu qu'on ne peut pas faire de décalage interne, et donc il faut, pour trouver deux correspondances, comparer les éléments de la plus petite liste depuis le début (forcément...) avec ceux de la plus grande depuis le deuxième, dans la limite des indices possibles.

Dans le fichier Python téléversé, il faut bien penser à mettre en première ligne et en commentaire le pseudo sur Codewars afin que je puisse suivre vos progrès!