

# RECURSIVITE

## Exercices - Enoncés

---

Dans ce recueil d'exercices, on doit OBLIGATOIREMENT fournir des versions récursives des codes demandés.

### Exercices d'entraînement

#### Exercice1 - Un calcul très classique

Ecrire une fonction Python qui calcule la somme des inverses des carrés des  $n$  premiers entiers naturels non nuls.

On pourra ensuite écrire un script plus complet qui, après le calcul précédent, évalue et affiche l'écart (en %) avec la limite de cette somme qui vaut  $\frac{\pi^2}{6}$  (rappel : le nombre  $\pi$  ne fait pas partie intégrante du cœur du langage Python. On importera donc `pi` via le module `math` : `from math import pi`).

#### Exercice 2 – Une fonction mystérieuse ?

```
def dk(L1,L2=[]):
    if L1 == []:
        return L2
    else:
        s = L1.pop(0)
        if s not in L2:
            L2.append(s)
        return dk(L1,L2)
```

Que renverra la fonction `dk` définie ci-dessus lorsqu'on l'appelle comme suit ?

```
dk([34,2,3,11,11,2,34,7,1,7,7,11,3,11])
```

### Exercice 3 – Le compte est-il bon ?

Ecrire une fonction Python « `compte_a` » qui reçoit comme argument une chaîne de caractères (éventuellement vide) et renvoie le nombre de « a » qu'elle contient.

La chaîne passée en argument peut contenir des lettres majuscules. On pourra souhaiter transformer la chaîne en une chaîne ne contenant que des minuscules (à vous de trouver la méthode qui réalise très bien cette transformation). Pourquoi n'est-il pas pertinent d'utiliser cette méthode dans votre fonction ?

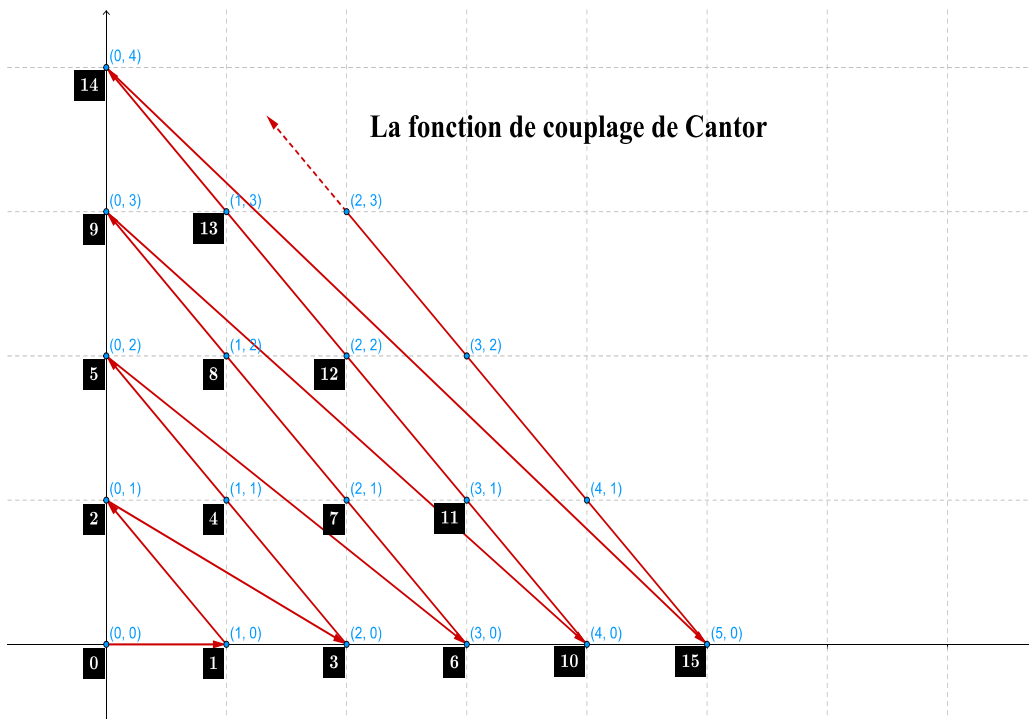
Vous pourrez tester votre fonction avec les chaînes suivantes : ' ', 'sphinx', 'abracadabra' et 'automorphisme'.

### Exercice 4 – Sens dessus dessous

Ecrire une fonction Python permettant de déterminer si une chaîne de caractères est ou non un palindrome (i.e. pouvant être lue indifféremment de la gauche vers la droite ou de la droite vers la gauche).

### Exercice 5 – Une bijection de $\mathbb{N} \times \mathbb{N}$ dans $\mathbb{N}$

Ci-après, une représentation de la fonction de couplage (notée  $f$ ) de Cantor (1845-1918) qui établit une bijection de  $\mathbb{N} \times \mathbb{N}$  dans  $\mathbb{N}$  (les couples d'entiers naturels sont représentés en bleu et les entiers associés en blanc sur fond noir). On a par exemple :  $f(1,1) = 4$  et  $f(1,3) = 13$ .



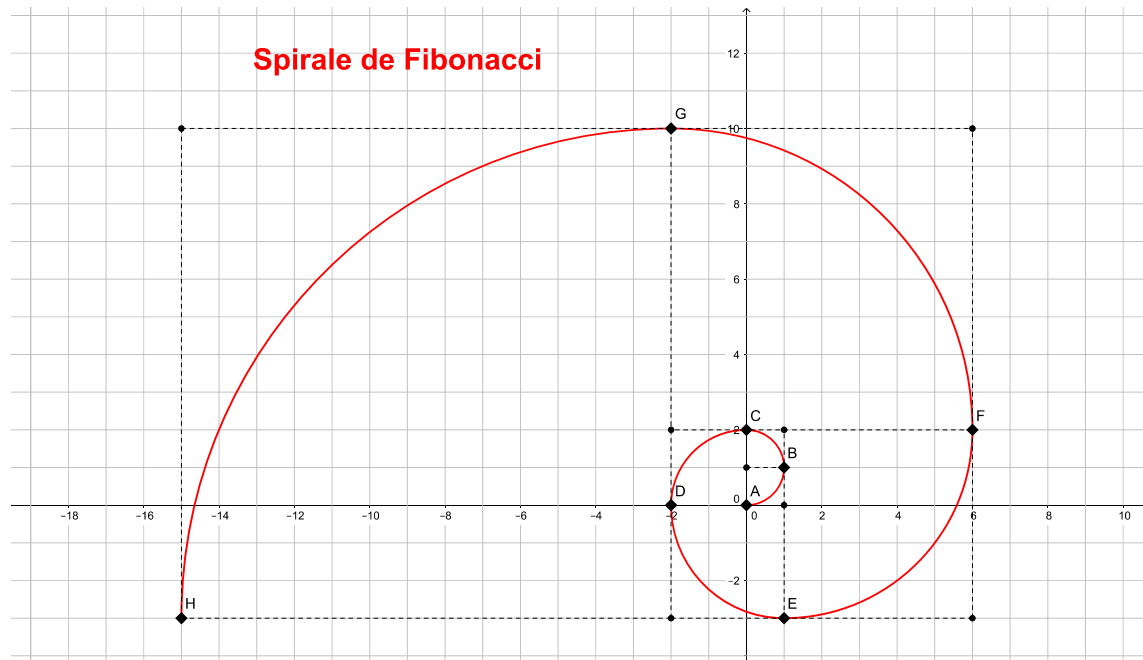
Ecrire une fonction qui permet de calculer  $f(x, y)$  pour tout couple  $(x, y)$  d'entiers naturels.

## Exercice 6 – Les tours ? C’est carré !

Ecrire une fonction Python permettant de placer  $n$  tours sur un échiquier  $n \times n$  de sorte qu’aucune ne soit en prise avec une autre (le programme devra déterminer toutes les configurations possibles et on affichera chacune des configurations. ATTENTION ! Il y en a beaucoup ! Et d’ailleurs, vous savez très bien combien ... ☺).

## Exercice 7 – La spirale de FIBONACCI

Un exercice (presque) sans énoncé :



Ecrire une fonction `SpiraleFibonacci` qui recevra en argument un entier naturel non nul  $n$  correspondant au nombre de quart de cercles à afficher et qui affichera... les  $n$  premiers quarts de cercle de la spirale de Fibonacci.

## Quelques suggestions pour aller plus loin ...

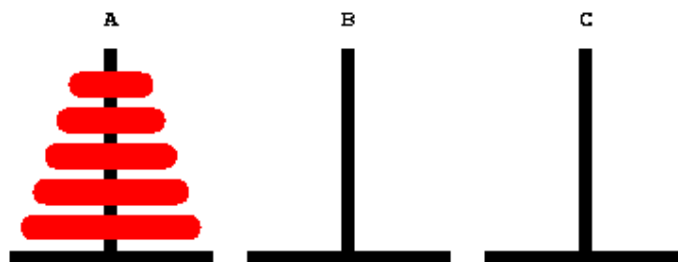
### Exercice 8 – Le pivot de GAUSS



Il est inutile de vous en rappeler le principe. En revanche, cette méthode peut assez aisément être implantée à l'aide d'une fonction récursive.

Comme on cherche fondamentalement à résoudre le système  $AX=B$ , l'appel initial à la fonction se fera en lui fournissant comme arguments A et B sous la forme de listes, la liste A servant à représenter une matrice carrée sous la forme d'une liste de listes (les lignes de la matrice à laquelle s'applique l'algorithme).

### Exercice 9 – Les tours de Hanoï



L'objectif de ce casse-tête est le suivant (cf. figure ci-dessus) : on doit déplacer les disques correspondant à la pile dans la position A vers la position C en utilisant la position intermédiaire B et en respectant les deux règles suivantes :

- On ne doit déplacer qu'un seul disque à la fois.
- A tout instant un disque donné doit se trouver en bas d'une pile ou sur un disque de diamètre plus important).

Le programme fournira, étape par étape, les déplacements à effectuer sous la forme, par exemple :  $A \rightarrow B$ . (ce qui signifie « déplacer le disque supérieur de la pile en A vers la pile (éventuelle) en B »).

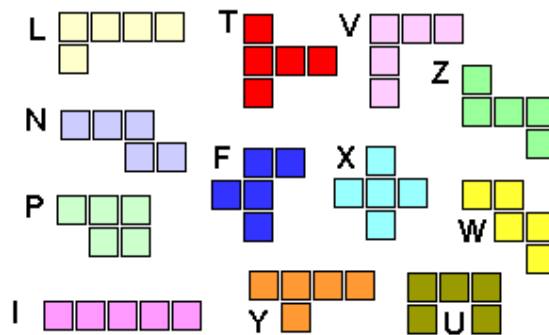
## Exercice 10 – Les dames ? C'est pas du gâteau !

Ecrire une fonction Python permettant de placer  $n$  dames sur un échiquier  $n \times n$  de sorte qu'aucune ne soit en prise avec une autre.

Le programme doit fournir toutes les solutions et donc disposer d'une fonction d'affichage (les solutions sont affichées au fur et à mesure de leur obtention).

## Exercice 11 – Les pentaminos

Les pentaminos, ce sont ces figures (cf. la figure ci-dessous, d'après wikipedia) que l'on peut obtenir à l'aide de 5 carrés élémentaires, tout carré devant être relié à au moins un autre carré par une arête.



L'un des problèmes (il y en a bien d'autres !) que l'on peut imaginer avec les pentaminos consiste à recouvrir un rectangle de dimension  $6 \times 10$  avec les douze pentaminos (chaque pentamino apparaissant une fois et une seule :

Il y a au total 2 339 solutions (en ne comptant pas les solutions obtenues par réflexion et rotation). Sauriez-vous les retrouver ?