I. Autour des nombres entiers

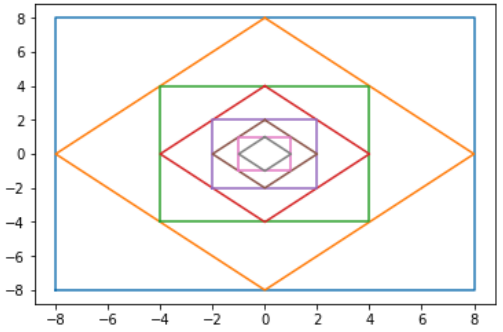
1. Écrire une fonction récursive calculant x^n pour $x \in \mathbb{Z}$ et $n \in \mathbb{N}$.
2. Écrire une fonction récursive somme(*n*) calculant la somme des *n* premiers entiers.
3. Écrire une fonction récursive nbchiffres(*n*) renvoyant le nombre de chiffres d'un entier.
4. Écrire une fonction récursive descente(n) affichant les entiers de 1 à *n* dans l'ordre décroissant.
5. Écrire une fonction récursive montee(n) affichant les entiers de 1 à *n* dans l'ordre croissant.
6. Écrire une fonction récursive rebond(n) qui donne le même résultat que descente(n) ; montee(n) sans utiliser ces fonctions.

II. Autour des listes

1. Écrire une fonction récursive affichant un à un les nombres d'une liste.
2. Écrire une fonction récursive recherchant le minimum dans une liste non vide.
3. Écrire une fonction récursive recherchant une valeur dans une liste triée par dichotomie, qui renvoie True si la valeur est présente dans la liste, False sinon.
4. (extrait d'un sujet du concours Centrale-Supélec) Écrire une fonction d'entête **def** somme(M): qui prend en paramètre une séquence imbriquée, de profondeur et de structure quelconques, dont tous les composants élémentaires sont des nombres, et calcule la somme de tous ces éléments.
   Par exemple : somme([[[1, 2], [3, 4, 5]], 6, [7, 8], 9]) → 45
   Indication : L'expression booléenne isinstance(x, numbers.Real) permet de tester si x est un nombre (après import numbers). Par exemple
   isinstance(1, numbers.Real) → True
   isinstance(2.3e4, numbers.Real) → True
   isinstance([1, 2, 3], numbers.Real) → False

III. Récursivité croisée

Écrire deux fonctions mutuellement récursives carreDroit(n) et carrePenche(n) telle que l'appel de carreDroit(8) affiche le dessin ci-contre.



On utilisera la fonction plot importée de matplotlib.pyplot pour dessiner les carrés.
Proposer une solution pour obtenir le même résultat avec une seule fonction récursive.

IV. Algorithme de Horner

Un polynôme P est représenté par la liste de ses coefficients.
Ainsi $X^3 – 3X^2 + 5X – 1$ est représenté par [1, -3, 5, -1].
En remarquant que $X^3 – 3X^2 + 5X – 1 = –1 + X(X^2 – 3X + 5)$, écrire une fonction récursive d'entête **def** horner(P, x): calculant P(x).

V. Composition d'un entier

1. Écrire une fonction récursive donnant la liste de toutes les compositions d'un entier *n* (décomposition de cet entier en une somme d'entiers strictement positifs tenant compte de l'ordre).
   Ainsi, si $n = 3$, cette fonction renverra [[1, 1, 1], [1, 2], [2, 1], [3]].
2. On s'intéresse maintenant à la liste des partitions d'un entier (décomposition de cet entier en une somme d'entiers strictement positifs sans tenir compte de l'ordre). Si $n = 3$, cette fonction renverra [[1, 1, 1], [1, 2], [3]].

Le nombre de partitions d'un entier naturel *p* en au plus *q* parties, noté $d(p, q)$, s'obtient par un algorithme récursif :

- $d(0, q) = 1$
- $d(p + 1, 0) = 0$
- $d(p, q) = d(p, p)$ pour $p < q$
- $d(p, q) = d(p – q, q) + d(p, q – 1)$

a. Écrire une fonction donnant le nombre de partitions d'un entier naturel *p*.
b. Écrire une fonction renvoyant la liste des partitions d'un entier.

Source **:** https://fr.wikipedia.org/wiki/Algorithme_récursif