La récursivité
Pour aller plus loin

1. Générer des listes

✍ Implémenter une fonction récursive **genere_liste_dec(n)** qui génère une liste décroissante depuis **n**.

```
>>> genere_liste_dec(5)
[5, 4, 3, 2, 1, 0]
```

✍ Implémenter une fonction récursive **genere_liste_cro(n)** qui génère une liste croissante jusqu'à **n**.

```
>>> genere_liste_cro(5)
[0, 1, 2, 3, 4, 5]
```

2. Nettoyer des listes

✍ Implémenter une fonction récursive qui nettoie une liste triée en éliminant les éléments identiques.

```
>>> nettoie([0, 1, 1, 2, 3, 4, 4, 5, 6, 6])
[0, 1, 2, 3, 4, 5, 6]
```

3. Convertir un nombre décimal en binaire

Pour cela, on utilisera la succession de divisions euclidiennes par 2.
Exemple : 29 en base 10. On effectue les divisions successives par 2 jusqu'à obtenir un quotient nul.

29	2
1	14

14	2
0	7

7	2
1	3

3	2
1	1

1	2
1	0

En lisant de droite à gauche les restes obtenus, on obtient l'écriture en base 2 de l'entier. Ici, $29_{10} = 1\ 1101_2$

✍ Implémenter la fonction récursive **dec_vers_bin()** qui prend un entier naturel comme argument et renvoie son écriture binaire sous la forme d'une chaîne de caractères.

```
>>> dec_vers_bin(8)
'1000'
```

✍ Renseigner le docstring et fournir un jeu de tests pertinent.

4. Convertir un nombre binaire en décimal

Si n est un entier naturel et a_0, a_1, …, a_p sont les chiffres de son écriture en base 2, on a :

$$n = 2^p a_p + 2^{p-1} a_{p-1} + \ldots + a_0$$

✍ En remarquant que $n = 2 \times (2^{p-1} a_{p-1} + \ldots + a_1) + a_0$ si $n \geq 2$, implémenter la fonction récursive **bin_vers_dec()** qui prend un chiffre binaire écrit sous forme de chaîne de caractère et renvoie son écriture décimal sous forme d'entier.

✍ Renseigner le docstring et fournir un jeu de tests pertinent.

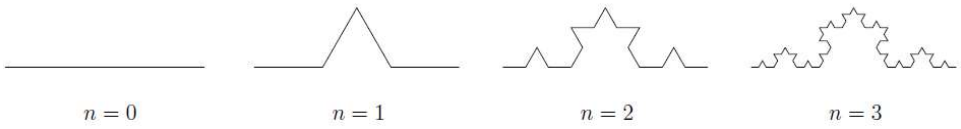
5. Flocon de Koch

Le **flocon de Koch** est l'une des premières courbes fractales à avoir été décrites. Elle a été inventée en 1904 par le mathématicien suédois **Helge von Koch**.

La courbe de Koch d'ordre n est définie de la manière suivante :

- si n = 0 c'est un segment de longueur 1 ;
- sinon :
  - on divise le segment de droite en trois segments de longueurs égales,
  - on construit un triangle équilatéral ayant pour base le segment médian de la première étape.

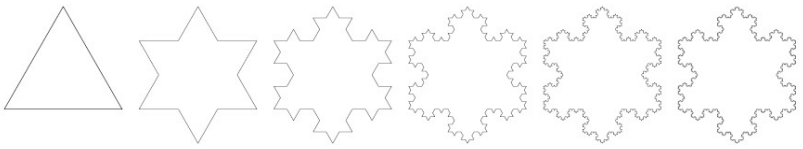
Courbes de Koch d'ordre n pour $0 \leq n \leq 3$:



L'objectif du problème est de dessiner cette courbe à l'aide de la bibliothèque **`Turtle`** à l'aide d'une fonction récursive.
Pour cela, on remarquera que la courbe de Von Koch d'ordre **`n + 1`** est constituée de 4 parties qui sont des courbes de Von Koch d'ordre **`n`** dont la taille a été divisée par 3.

✍ Implémenter une fonction récursive **`courbe(n, long)`** qui dessine cette courbe en utilisant **`Turtle`** avec **`n`** l'ordre de la courbe et **`long`** la longueur du segment à l'ordre 0.

✍ En utilisant la fonction précédente, écrire une fonction **`flocon(n, long)`** dessinant le flocon de Von Koch d'ordre **`n`**. Ci-dessous les flocons de l'ordre 0 à l'ordre 5 :



6. Palindrome

Un palindrome est un mot qui se lit de la même manière dans les deux sens.
« kayak » et « laval » sont des palindromes.

✍ Implémenter la fonction récursive **`palindrome()`** qui prend un chaîne de caractère comme argument et renvoie vrai si c'est la chaîne est un palindrome sinon faux.

✍ Renseigner le docstring et fournir un jeu de tests pertinent.