



## Exercice 1 — Une suite

La suite  $(t_n)$  est définie par  $t_0 = 1$  et pour tout  $n \in \mathbb{N}^*$ ,  $t_{2n} = t_n$  et  $t_{2n+1} = 1 - t_n$

1. Calculer les 10 premiers termes de cette suite.
2. Calculer  $t_{2024}$ .
3. Existe-t-il une valeur de  $n$  telle que  $t_n = t_{n+1} = t_{n+2}$ ? Justifier.

## Exercice 2 — L-system

Un L-system, ou un système de Lindenmayer, est un système de grammaire formelle, inventé en 1968 par le biologiste hongrois Aristid Lindenmayer. Un L-system modélise le processus de développement et de prolifération de plantes ou de bactéries. L'idée est d'associer une instruction de la tortue LOGO (ou la tortue Python) à chaque chiffre de  $t_n$ , (terme de rang  $n$  de la suite définie en partie 1).

On décide que 0 signifie « avancer de 1 unité » et 1 « tourner à gauche de  $60^\circ$  (sans avancer) ».

Compléter le programme suivant et le tester (pour des valeurs de  $n$  de l'ordre de 10 000, réduire le déplacement de la tortue pour voir le dessin en entier).



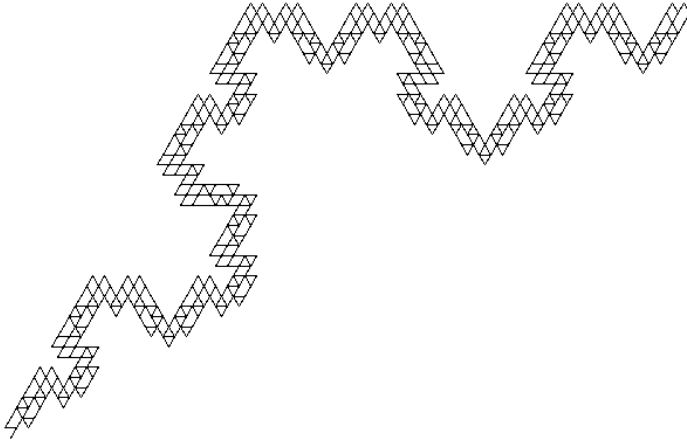
---

```
1  -*- coding:utf-8 -*-
2  # pyhton3
3
4  import turtle as tl
5
6  # je ne cautionne pas... je préfère écrire une fonction
7  # mais c'est le type d'instruction qu'on trouve au BAC...
8  n = int(input("rang de la suite "))
9  # tm est la liste des valeurs de la suite
10 tm = [1] # la valeur de t_0
11 for k in range(1, n): # jusqu'à tm_(n-1)
12     if <...> : # le rang k est pair
13         tm.append(tm[k // 2]) # append = ajoute en fin de
14         ↔liste
15     else: # le rang est impair
16         tm.append(<...>)
17
18 # préparation de la tortue
19 tl.penup()
20 tl.setposition(-300,-250)
21 tl.pendown()
22 tl.speed(0)
23 tl.hideturtle()
24 # dessin fait par la tortue lors du parcours de la liste
25 for k in tm:
26     if k == 0:
27         tl.forward(5) # mouvement de la tortue
28     else:
29         tl.left(60) # mouvement de la tortue
30 tl.mainloop()
```

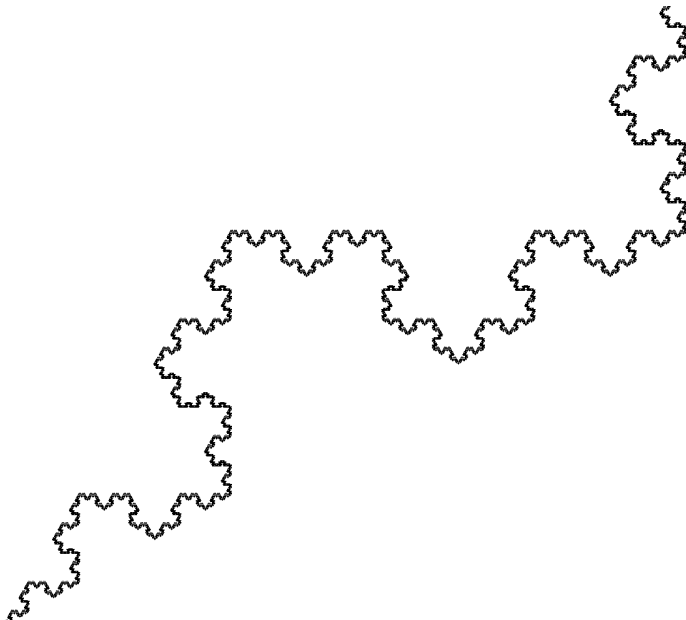
---



avec  $n = 2000$  et `forward(10)`



avec  $n = 500000$  et `forward(2)`

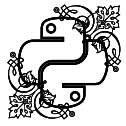




## le flocon de Von Koch avec les L-System

---

```
1  # -*- coding:utf-8 -*-
2  # python3
3  #
4  # par défaut on par d'un segment (mot="F")
5  # chaque segment (représenté par F)
6  # est transformé par le motif.
7  # Par défaut motif="FLFLLLLFLF" (L = gauche 60°)
8
9  import turtle as tl
10
11 def Lsystem(n, mot="F", motif="FLFLLLLFLF"):
12     if n > 5:
13         print("Attention au temps d'exécution")
14         choix = input("Continuer ? (o/n) ")
15         if choix not in ["o", "O", "y", "Y"]:
16             return n
17
18     for _ in range(n):
19         m_tmp = ""
20         for c in mot:
21             if c == "F":
22                 m_tmp = m_tmp + motif
23             else:
24                 m_tmp = m_tmp + c
25         mot = m_tmp
26
27     # on peut plus simplement écrire
28     # for _ in range(n):
29     #     mot = mot.replace("F", motif)
30
31     # préparation de la tortue
```



```
32     tl.penup()
33     tl.setposition(-300,-250) # coordonnées du début
34     tl.pendown()
35     tl.speed(0)
36     tl.hideturtle()
37     pas = max(1, int((2/3)**n * 40)) # choix arbitraire
38     # la tortue lit la liste pour se déplacer
39     for c in mot:
40         if c == "F":
41             tl.forward(pas)
42         else:
43             tl.left(60)
44     tl.mainloop()
```

---

Tester d'autres mot avec d'autres motifs :

par exemple, le flocon s'obtient avec `mot="FLLLLFLLLLF"` et modifier les coordonnées de départ.