La courbe de Sierpiński

Wacław Sierpiński décrit le triangle de Sierpiński (aussi appelé le joint de Sierpiński ou le tamis de Sierpiński) en 1915. La courbe de pointe de flèche de Sierpiński permet de réaliser ce triangle assez simplement. En fait on retrouve ce motif dans des motifs décoratifs d'églises ou de tapis plusieurs siècle avant le travail de Sierpiński.

Construction des premières étapes de la courbe

1. On part d'un segment (de longueur arbitraire `c`)

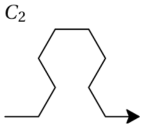
C_0

2. On remplace ce segment par trois segments de longueur `c/2` :
Le premier fait un angle de 60° avec le segment remplacé, le second lui est parallèle est le troisième forme un angle de -60° avec le segment remplacé.

C_1

3. On recommence pour l'étape 3 avec la contrainte supplémentaire suivante : à chaque remplacement d'un segment, l'orientation de 60° alterne : la rotation s'effectue dans le sens opposé pour les segments impaires et paires :

C_2



4. Dessinez l'étape 4 de cette courbe puis appelez moi pour vérifier.

Le programme Python

1. Voici le code pour tracer la première étape, testez le :

```
from turtle import * #importation du module turtle

c=500 #longueur du segment de base
forward(c) #avance de c
done() #termine les deplacements de la tortue
```

2. Complétez le code suivant pour obtenir le tracé de la seconde étape.

```
k = 1 #coefficient pour choisir l orientation
c = 500
left(k*60)
forward(c/2)
right(k*60)

done()
```

3. Écrire une fonction récursive `sierp(n, c, k)` qui trace la n-ième étape de la courbe de Sierpinski.
`n` est donc l'étape cherchée et `c` la longueur d'un segment à cette étape (pour l'étape 0 par exemple `c` vaut 500 et pour l'étape suivante `c` vaut 250) et `k` définit l'orientation (pour l'étape 0 : `k` vaut 1, à l'étape 1 : `k` vaut -1 puis à l'étape 2 : `k` vaut 1).