L-Systèmes

- Les systèmes de Lindenmayer, appelés aussi L-Systèmes, ont été imaginés par le biologiste Aristid Lindenmayer(1925-1989) et modélisent le processus de développement et de prolifération de plantes.
- Le concept central des L-Systèmes est la représentation d'une plante par une chaîne de caractères. Cela permet de modéliser son évolution, voire sa destruction par des agents pathogènes, au moyen de règles de transformations de ces caractères.

- On utilisera ici le module `turtle` pour représenter un L-Système.

1. Un alphabet pour coder une figure.

Définissons une figure f par la donnée d'un triplet contenant :

- Une longueur L.
- Un pas de rotation a, donné en degrés.
- Un mot, appelé motif, utilisant les caractères de l'ensemble {F ; + ; - } avec comme convention :
  - "F" : avancer de L
  - "+" : tourner à gauche de a degrés
  - "-" : tourner à droite de a degrés

Par exemple, la figure $f(50; 90; F+F+F+F)$ représente un carré dé 50 unités de côté.

Exercice 1 :

1. Que représente la figure $f(20; 60; F++F++F)$?

Réponse :

1. A quel figure f peut correspondre le dessin ci-dessous ?

Réponse :

Exercice 2 :

Ecrire une fonction `dessiner(unite, angle, motif)` qui :

- reçoit en paramètres :
  - `unite` : un nombre représentant la longueur L.
  - `angle` : un nombre représentant l'angle a de rotation.
  - `motif` : le motif m de la figure sous forme d'une chaîne de caractères
- affiche le dessin de la figure $f(L, a, m)$ avec le module `turtle`

In [1]:

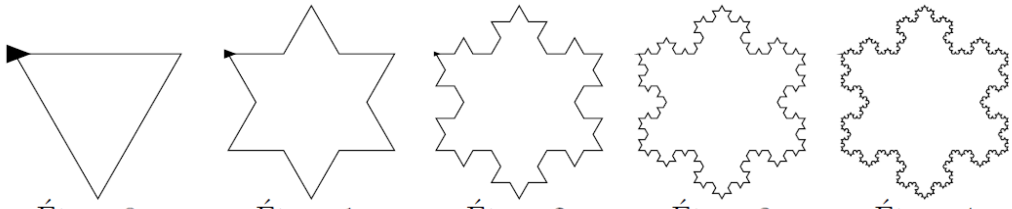
```
from turtle import *

def dessiner(unite, angle, motif) :
    pass
```

In [2]:

```
#tests
dessiner(50,60,'F++F++F')
mainloop()
```

2. Premiers L-Systèmes

L'intérêt des L-Systèmes est de permettre de décrire simplement l'évolution d'une figure. Prenons l'exemple du flocon de Von Koch (l'échelle d'une image à l'autre a été ajustée pour une meilleure visibilité) :



Étape 0 Étape 1 Étape 2 Étape 3 Étape 4

En choisissant un pas d'angle de rotation de 60° , les motifs des deux premières figures sont :

Étape 0 : `F -- F -- F`

Étape 1 : `F+F--F+F -- F+F--F+F -- F+F--F+F`

A chaque étape, chaque lettre F représentant un segment est remplacé par le motif $F+F--F+F$.
Un L-système est la donnée d'un *axiome* (motif de départ) et d'une *règle* ou d'un ensemble de règles.

Dans notre exemple, l'axiome est $F--F--F$ et la règle $F+F--F+F$.

Exercice 3 :

Ecrire une fonction `suivant(motif,regle)` qui :

- reçoit en paramètres deux chaînes de caractères :
  - motif : le motif de la figure à une étape donnée
  - regle : la regle d'évolution de la figure
- retourne en sortie une chaîne de caractères représentant la figure à l'étape suivante.

Par exemple : `suivant('F--F--F','F+F--F+F')` renvoie `'F+F--F+F--F+F--F+F--F+F--F+F'`

In [4]:

```
def suivant(motif,regle):
    res=''




    return res
```

In [6]:

```
#
dessiner(50,60,suivant('F--F--F','F+F--F+F'))
mainloop()
suivant('F--F--F','F+F--F+F')
```

Out[6]:

```
''
```

Exercice 4 :

Programmer la fonction `evolution(axiome,regle,etape)` qui :

- reçoit en entrée :
  - axiome : une chaîne de caractères représentant le motif de départ
  - regle : une chaîne de caractères indiquant la règle d'évolution
  - etape : un entier indiquant le numéro de l'étape à calculer
- retourne en sortie une chaîne de caractères représentant la figure à l'étape demandée.

Par exemple : `evolution('F','F+' ,4)` renvoie `'F++++'`

In [7]:

```
def evolution(axiome, regle,etape):
    pass
```

In [8]:

```
evolution('F','F+',4)
```

Exercice 5 :

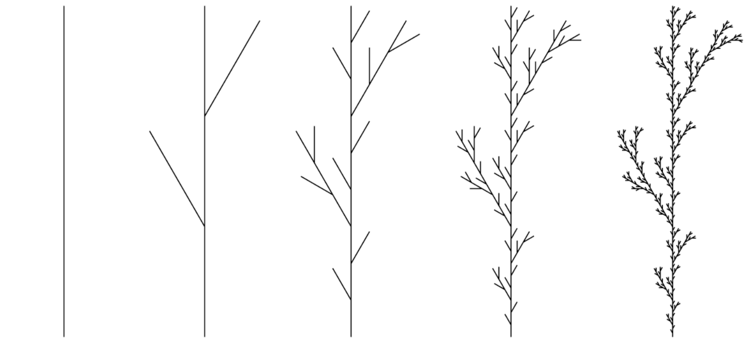
En utilisant les foncitons précédentes, dessiner le flocon de Von Koch à l'étape 4

In [10]:

```
speed(0)

#Von Koch


mainloop()
```

3. Gestions des ramifications

On souhaite à présent représenter une plante à l'aide d'un L-Système. Par exemple, avec une règle préalablement choisie et partant d'une branche F, on peut obtenir les images suivantes où le pas de rotation est de 20° (l'échelle est ajustée d'une image à l'autre et l'azimut de départ initialisée à 90° pour un rendu plus réaliste) :



Étape 0 Étape 1 Étape 2 Étape 3 Étape 4

On se propose d'ajouter deux nouveaux symboles à l'alphabet des L-Systèmes :

- " [" : place l'état de la tortue (coordonnées et orientation) en tête d'une pile.
- "] " : dépile la dernière position de la tortue et replace la tortue à cet endroit (sans effectuer de tracé).

Par exemple, le motif de l'étape 1 est alors décrit par la chaîne "F[+F]F[-F]F"

Exercice 6 :

Que représente la figure $f(2, 90, F[-F[+F]-F]F)$:

Réponse :

Exercice 7 :

Réécrire la fonction `dessiner(unite,angle,motif,azimut)` pour qu'elle tienne compte de ces deux nouveaux symboles. Elle prend en paramètres :

- `unite` : la longueur
- `angle` : le pas de rotation a en degrés
- `motif` : une chaîne de caractèresreprésentant le motif de la figure
- `azimut` : l'azimut initial du tracé (il est égal à 0 par défaut)

Indications :

- La pile qui mémorise l'état de la tortue sera géréee par une liste et par les méthodes `append` et `pop` .
- Quelques commandes `turtle` :
  - `heading()` : renvoie le cap en degrés de la tortue
  - `position()` : renvoie un tuple qui contient les coordonnées cartésiennes de la tortue.
  - `setheading(cap)` : oriente la tortue au cap passé en argument, en degrés
  - `goto(x,y)` : déplace la tortue aux coordonnées cartésiennes passées en arguments
  - `up()` : relève le stylo
  - `down()` : abaisse le stylo

In [14]:

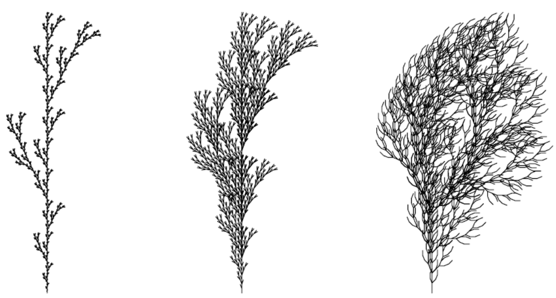
```
def dessiner(unite, angle, motif, azimut) :
    pile=[]
    setheading(azimut)
    pass
```

In [16]:

```
#tests
hideturtle()
dessiner(20,90,'F[-F[+F]-F]F',45)
mainloop()
```

Exercice 8 :

- L'image ci-contre est extraite de l'ouvrage d'Aristid Lindenmayer "Algorithmic Beauty Of Plants".
- En utilisant les différentes fonctions écrites précédemment, dessiner les trois exemples présentés sur cette image.



a
n=5 , δ=25.7°
F
F →F [+F] F [-F] F

b
n=5 , δ=20°
F
F →F [+F] F [-F] [F]

c
n=4 , δ=22.5°
F
F →FF- [-F+F+F] + [+F-F-F]

In [18]:

```
#initialisaiton turtle
tracer(120)
up()
goto(0,-200)
hideturtle()


#Exemples


mainloop()
```

4. En savoir plus

Il ne s'agissait ici que d'un bref aperçu des L-Systèmes. Cette modélisation est très puissante et permet, au delà de l'intérêt scientifique, de pousser le réalisme visuel obtenu à des niveaux très impressionnants :

- Vidéo générée avec le logiciel Houdini utilisant les L-Systèmes : https://vimeo.com/356351056 (https://vimeo.com/356351056)
- Page wikipédia sur les L-sytèmes : https://fr.wikipedia.org/wiki/L-Syst%C3%A8me (https://fr.wikipedia.org/wiki/L-Syst%C3%A8me)