Exercice 1

1. Ecrire une classe *Point* ayant pour attributs :*abscisse* et *ordonnee*.

```
from math import *
class Point():
def __init__(self,abscisse,ordonnee):
    ..............................
    ..............................
```

2. Créer trois instances A(10,10) , B(20,20) et C(30,15) de l'objet *Point*.
3. Ajouter à la définition de la classe *Point* , la méthode spéciale __str__ pour afficher les coordonnées de ces points dans le shell.
4. Ecrire une fonction *translater(P,a,b)* qui renvoie un objet *Point* image du point P par la translation de vecteur $\overrightarrow{U}(a,b)$.
5. Ecrire une fonction *distance* qui calcule la distance euclidienne entre deux points (objets) A et B.

Exercice 2

1. Ecrire une classe *Triangle* ayant pour attributs trois points (objets) A, B et C
2. Ecrire une méthode *perimetre* qui renvoie le périmètre d'un triangle.
3. Ecrire une fonction *translate_ triangle(T,a,b)* qui renvoie le triangle image d'un triangle T par la translation de vecteur (a,b).

Exercice 3(le module pylab)
import pylab
On utilise le module pylab pour tracer des figures géométriques.Par exemple:

L'instruction pylab.plot(x,y,'bo') permet de placer un point de coordonnées (x,y).

l'instruction pylab.plot([x1,y1],[x2,y2]) permet de tracer un segment reliant deux points de coordonnées (x1,y1) et (x2,y2).

L'instruction pylab.Polygon([(xa,ya),(xb,yb),(xc,yc)],fill=False) permet de tracer le triangle dont les sommets ont pour coordonnées (xa,ya),(xb,yb),(xc,yc).

L'instruction pylab.Circle([x , y] , radius=r , fill =False) permet de tracer le cercle de centre le point de coordonnées (x,y) et de rayon r.

Dans un programme Python , on importe le module *pylab* pour pouvoir afficher les dessins ci-dessus.

```
import pylab
F=pylab.gca ( )

#On écrit les instructions pour tracer des figures géométriques
.........................
.........................
.........................
pylab.axis( 'equal')
pylab.axis('off')
pylab.show ( )  #Pour permettre l'affichage
```

Exercice 4

On veut construire un module 'geometrie.py' qui contient les classes objets *Point* ,*triangle* ainsi que des fonctions qui traduisent des tranformations géométriques telles que : *translation* , *homothétie* , *rotation*.

On donne par exemple ci-dessous les fonctions *homothétie* , *rotation*

```
def homothetie(A,Omega,k):
    '''
    A : le point dont on veut calculer l'image
    Omega : le centre de l'homothetie
    k : rapport
    '''
    x=A.abscisse
    y=A.ordonnee
    w1=Omega.abscisse
    w2=Omega.ordonnee
    return Point(k*(x-w1)+w1 , k*(y-w2) + w2)

def rotation(A,Omega,teta):
    '''
     A : le point dont on veut calculer l'image
     Omega : le centre de la rotation
     teta : angle de la rotation
    '''
 # la rotation renvoie l'image du  point P(x,y)

    x=A.abscisse
    y=A.ordonnee
    w1=Omega.abscisse
    w2=Omega.ordonnee
    a , b =cos(teta)  , sin(teta)
    return Point(a*(x-w1)-b*(y-w2)+w1,b*(x-w1)+a*(y-w2)+w2)
```

On veut également construire une classe *polygone_ regulier* qui prend pour paramètres , un centre de type **Point** , un sommet quelconque du polygone (de type **Point**) et un enier n égal au nombre de sommets du polygone.

```
class polygone_regulier:

    def __init__(self,centre,sommet,n):

        '''
        entree : -Point centre est le point centre du polygone
        sommet : -Point est un sommet du polygone
        n: -int est le nombre de sommets du polygone
       '''
       self.nbrsommets=n
       self.centre=centre
       #la liste des sommets contient sommet
       self.Sommets=[sommet]
       A=sommet
      #on construit les autres sommets par rotation
       for i in range(n-1):
           image=rotation(A,centre,2*pi/n)
           self.Sommets.append(image)
           A=image
```

Tester ces fonctions pour construire un triangle équilatéral , un carré ou un hexagone régulier.

Exercice 5

Dans l'exercice 4 , on a écrit les instructions permettant de tracer des figures géométriques.
Créer un fichier Python contenant ces instructions.
Importez-y le module 'geometrie' écrit dans l'exercice 4.

1. Ecrire une fonction *triangle_ homothetique*(T,centre,k) qui prend pour paramètre un objet triangle T , un point *centre* , centre de l'homothétie et un paramètre de type 'float' k (rapport de l'homothétie) et qui renvoie l'image du triangle par l'homothétie.

2. Ecrire de même une fonction *poly_ homothetique*(poly,centre_hom,k) qui prend pour paramètre un objet *poly* (objet *polygone_ regulier*) , un point *centre_ hom centre* , centre de l'homothétie et un paramètre de type 'float' k (rapport de l'homothétie) et qui renvoie l'image du polygone régulier *poly* par l'homothétie.

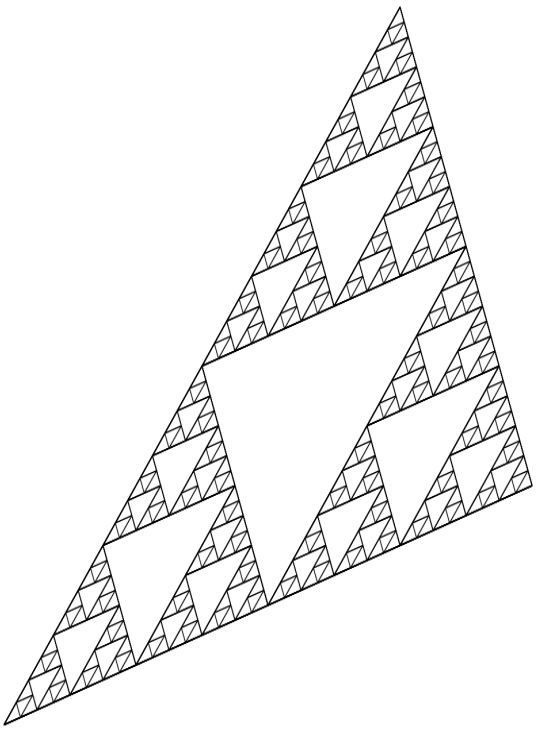
3. Triangle de Sierpinski:

   Construire un triangle (objet *triangle*) de sommets trois points A,B et C et tracer-le à l'aide du module *pylab*.
   Utiliser la fonction *triangle_ homothetique* pour construire l'image du triangle ABC par chacune des homothéties suivantes:

   - homothétie de centre A , de rapport 1/2.
   - homothétie de centre B , de rapport 1/2.
   - homothétie de centre C , de rapport 1/2.

   Recommencer plusieurs fois le même algorithme de construction sur chacun des triangles ainsi construits.
   Vous obtiendrez le triangle de Sierpinski ci-dessous.

   On écrira la fonction récursive *triangle_ sierpinski*(T) qui permet d'obtenir un tel triangle .



4. Construire le triangle de Sierpinski à partir d'un triangle équilatéral (polygone régulier de 3 côtés)

Exercice 6

Un carré est la réunion de 9 carrés.
Si on supprime le carré central en gardant les 8 autres , on obtient la figure 1.

Les 8 carrés sont obtenus en prenant les images du carré initial par les homothéties de rapport 1/3 et de centres les 4 sommets et les 4 milieux des côtés (fig 2))

Si on recommence plusieurs fois le même algorithme de construction sur chacun des carrés ainsi construits, on obtient la carpette de Sierpinski (fig 3).

Ecrire la fonction récursive *carpette_ sierpinski*(T) qui permet d'obtenir une telle figure.

fig 1

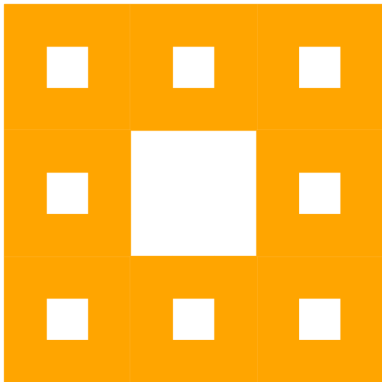


fig 2

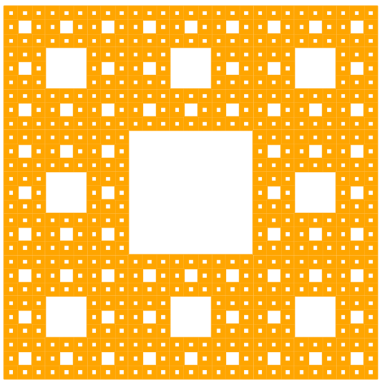


fig3

Exercice 7

On part d'un segment de longueur l et on le remplace par quatre segments de longueur $\frac{1}{3}l$ chacun.

On réitère le procédé sur chacun des segments ainsi construits.

En utilisant des homothéties et des rotations appropriées , sauriez-vous construire la courbe de Koch (fig 3)?

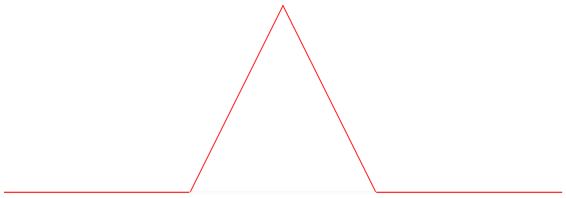


fig 1

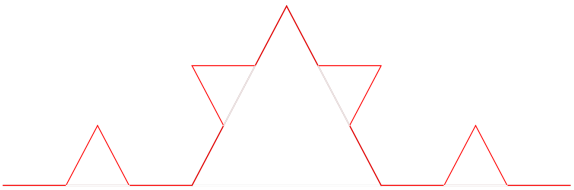


fig 2

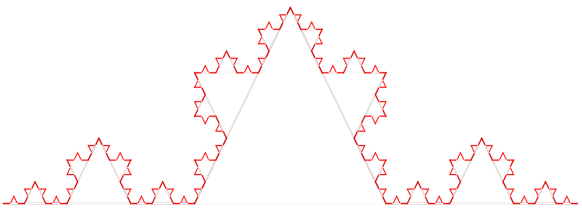


fig3