

Dernière mise à jour	Informatique	Claire GAUDY - Denis DEFAUCHY
12/01/2023	5 - Fonction récursives	TD 5-3 - Figures récursives

# Informatique

## 5

# Fonctions récursives

***TD 5-3***

***Figures récursives***

Dernière mise à jour	Informatique	Claire GAUDY - Denis DEFAUCHY
12/01/2023	5 - Fonction récursives	TD 5-3 - Figures récursives

## Exercice 1: Figures simples

Rappel :

```
>>> n=5

>>> '*'*n
'*****'
```

**Question 1:** Ecrire une fonction récursive `triangle_bas(n)` permettant d'afficher un triangle pointe en bas dont la 1<sup>ère</sup> ligne contient n étoiles

Vérifier :

```
>>> triangle_bas(5)
*****
****
***
**
*
```

**Question 2:** Ecrire une fonction récursive `triangle_haut(n)` permettant d'afficher un triangle pointe en haut dont la dernière ligne contient n étoiles

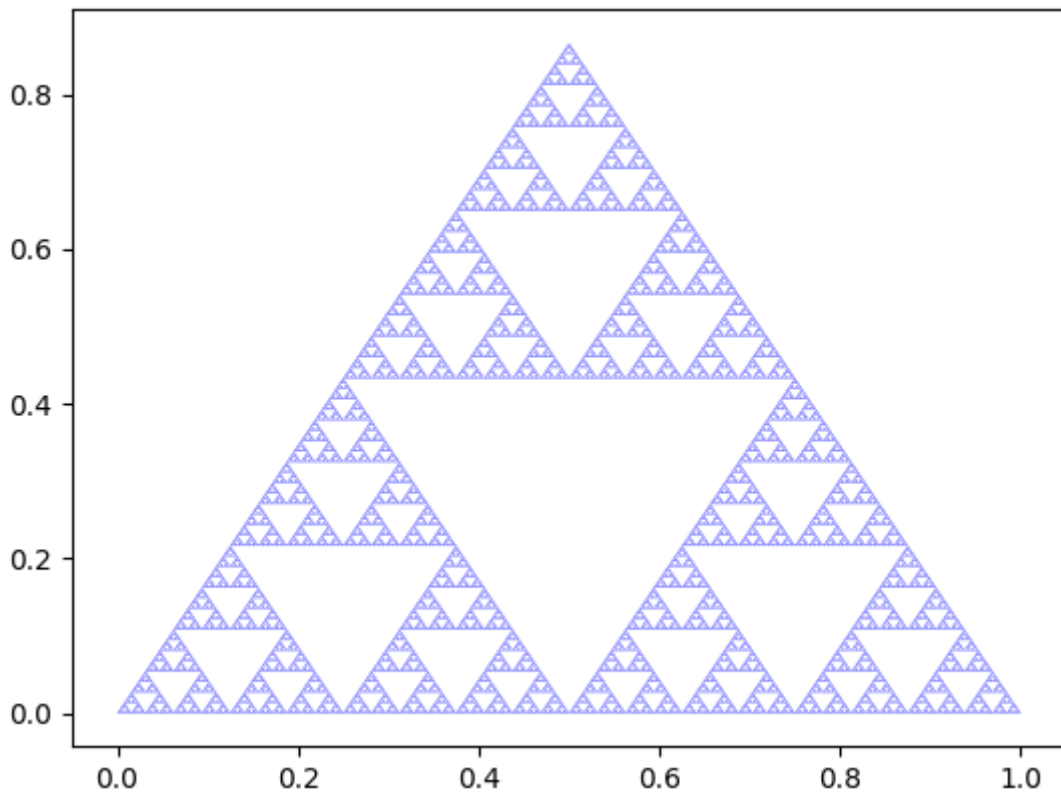
Vérifier :

```
>>> triangle_haut(5)
*
**
***
****
*****
```

Dernière mise à jour	Informatique	Claire GAUDY - Denis DEFAUCHY
12/01/2023	5 - Fonction récursives	TD 5-3 - Figures récursives

## Exercice 2: Triangle de Sierpinski

**Question 1:** Ecrire une fonction `Milieu(M1,M2)` qui, étant données deux listes de coordonnées de deux sous la forme  $[X_i, Y_i]$ , renvoie une liste des coordonnées du milieu du segment  $[M1, M2]$



Le code suivant permet de tracer sur une figure un polygone dont on définit les  $n$  sommets en argument sous la forme d'une liste de couples de coordonnées  $[[X_A, Y_A], [X_B, Y_B], \dots, [X_n, Y_n]]$  :

```
from matplotlib import pyplot as plt
plt.close('all')
```

```
def Polygon(Liste_Points) :
    X = []
    Y = []
    for Point in Liste_Points :
        X.append(Point[0])
        Y.append(Point[1])
    plt.fill(X,Y,'k')
    plt.show()
```

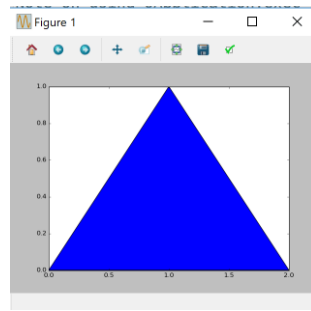
[LIEN](#)

```
A = [0,0]
B = [1,1]
C = [2,0]
```

```
Polygon([A,B,C])
```

Dernière mise à jour	Informatique	Claire GAUDY - Denis DEFAUCHY
12/01/2023	5 - Fonction récursives	TD 5-3 - Figures récursives

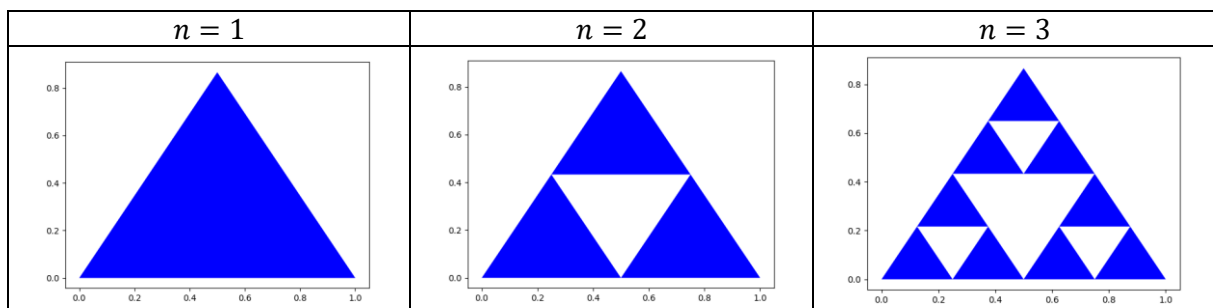
L'exécution de ce code renvoie :



**Question 2:** Ecrire une fonction récursive `Sierpinski(n,A,B,C)` traçant le résultat affiché ci-dessous (triangles équilatéraux)

**Question 3:** Préciser la complexité de cette fonction

**Question 4:** Ecrire une fonction récursive `Sierpinski(n)` réalisant le même travail



Remarques :

- Si tout est bleu, vous avez oublié d'enlever le triangle du code que je vous ai fourni en exemple !
- Un peu d'aide ?
  - o Il faut que la fonction s'appelle 3 fois à chaque étape
  - o Les triangles bleus ne sont tracés que si  $n = 1$
- On impose de dessiner des triangles colorés à chaque étape (pas de triangles blanc). Au passage, si vous y réfléchissez, il y aurait en faisant des triangles blancs 3 appels à chaque étape aussi, mais un dessin au lieu de 3 dans chacun des 3 sous triangles
- Si cela vous amuse, modifiez la fonction `Polygon` en ajoutant ces 3 lignes avant `plt.show()` afin de voir se construire les différents triangles lors de l'exécution de `Sierpinski` :

```
plt.xlim([0,1])
plt.ylim([0,1])
plt.pause(0.5)
```

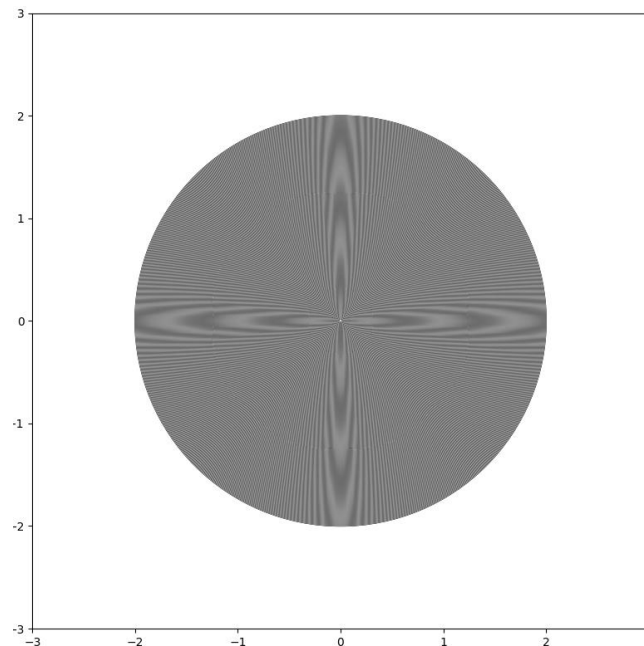
BUG Matplotlib : Sur certains ordinateurs, soit il faut fermer les figures pour que la suivante apparaisse, soit chaque triangle est créé sur une nouvelle figure. Dans ce cas, la solution est simple :

- Enlever « `plt.show()` » de la fonction `Polygon`
- Ajouter un seul « `plt.show()` » après l'appel de la fonction `Sierpinski`

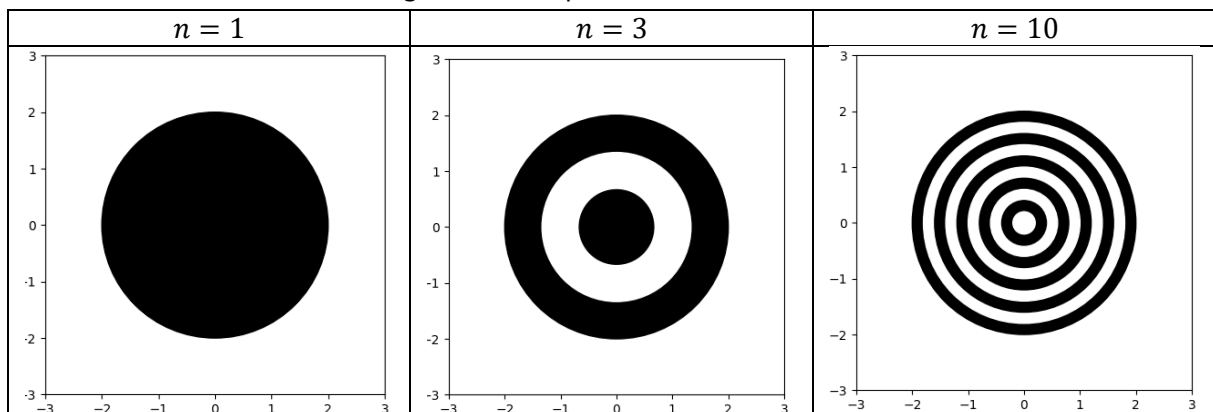
**Ne réaliser les exercices suivants que si vous avez de l'avance**

Dernière mise à jour	Informatique	Claire GAUDY - Denis DEFAUCHY
12/01/2023	5 - Fonction récursives	TD 5-3 - Figures récursives

### Exercice 3: Cercles concentriques



Nous allons mettre en place un algorithme qui va réaliser une figure récursive où l'on affiche des cercles concentriques de couleurs successivement noire puis blanche où le rayon diminue avec un pas constant. Voici ci-dessous les images à obtenir pour différents cas :



Tous les cercles sont centrés sur l'origine de coordonnées (0,0). Le premier cercle est noir et de rayon 2. Le pas entre deux rayons est calculé en fonction de  $n$  tel que  $dr = 2/n$

$N$  définit le nombre de cercles à afficher :

- $n=1$ 
  - On affiche un seul cercle de rayon 2
- $n=2$ 
  - Le premier cercle est noir de rayon 2
  - Le second est blanc de rayon 1
- $n=3$ 
  - Le premier cercle est noir de rayon 2
  - Le second est blanc de rayon 1,34
  - Le troisième est noir de rayon 0,68

Sous python, noir c'est 'k', et blanc c'est 'w'.

Dernière mise à jour	Informatique	Claire GAUDY - Denis DEFAUCHY
12/01/2023	5 - Fonction récursives	TD 5-3 - Figures récursives

Voici un code partiel qu'il vous faudra compléter :

```
import matplotlib.pyplot as plt
plt.close('all')

def Cercle(r,c):
    x = 0
    y = 0
    Circle = plt.Circle((x,y),r,color=c)
    plt.gca().add_artist(Circle)

def Affiche(x_min,x_max,y_min,y_max):
    plt.xlim(x_min,x_max)
    plt.ylim(y_min,y_max)
    plt.show()
    plt.gca().set_aspect('equal', adjustable='box')

Cercle(2,'k')
Cercle(1,'w')
Affiche(-3,3,-3,3)
```

Quelques précisions :

- L'appel de la fonction `Cercle(r,c)` ajoute à la figure en cours un cercle centré sur l'origine, de rayon `r`, et de couleur `c` (`c` doit être soit `'k'`, soit `'w'`)
- L'appel de la fonction `Affiche(-3,3,-3,3)` affiche l'image dans l'intervalle `x=-3...3` et `y=-3...3`.

### Question 1: Créer une fonction `Cercle(n)` permettant de réaliser les figures proposées

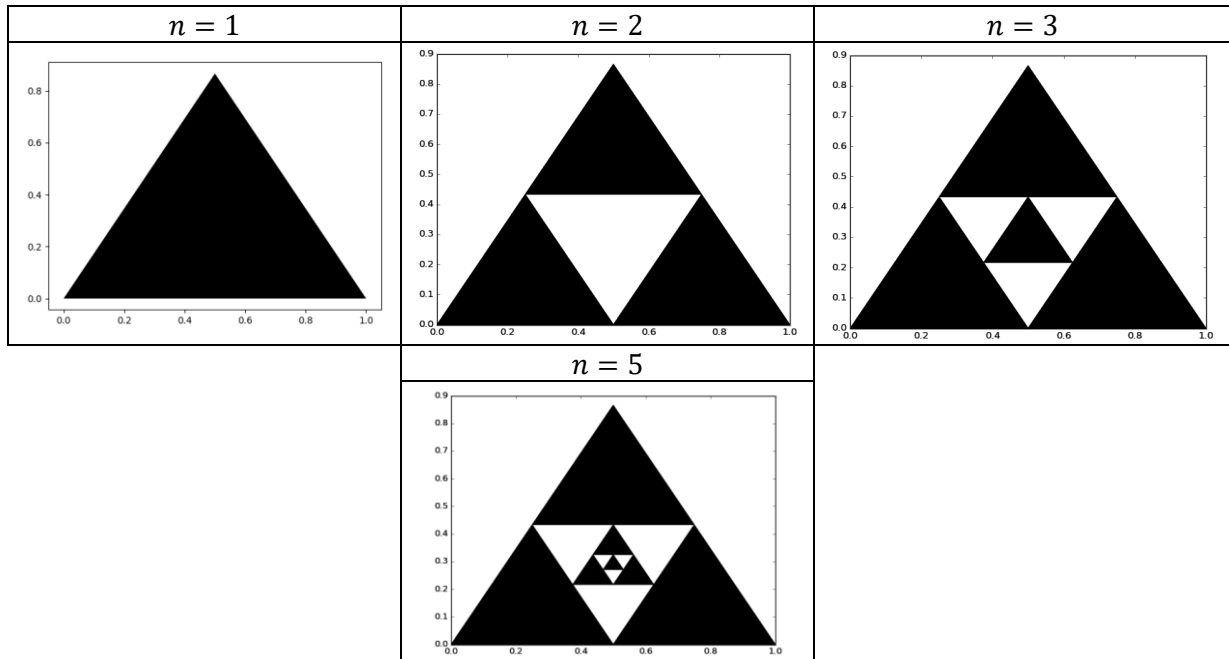
Remarques :

- On veillera à afficher les cercles au bon moment... En effet, si le plus petit est affiché en premier, il sera recouvert par le second, et ainsi de suite... Si vous n'y prêtez pas attention, à la fin, il n'y en aura qu'un d'affiché !
- On pensera à créer une fonction locale `rec` dont les arguments seront à choisir

Dernière mise à jour	Informatique	Claire GAUDY - Denis DEFAUCHY
12/01/2023	5 - Fonction récursives	TD 5-3 - Figures récursives

## Exercice 4: Triangles infinis

Voici maintenant les images suivantes :



**Question 1: Ecrire une fonction récursive `Triangles_Infinis` traçant le résultat affiché ci-dessus (triangles équilatéraux)**

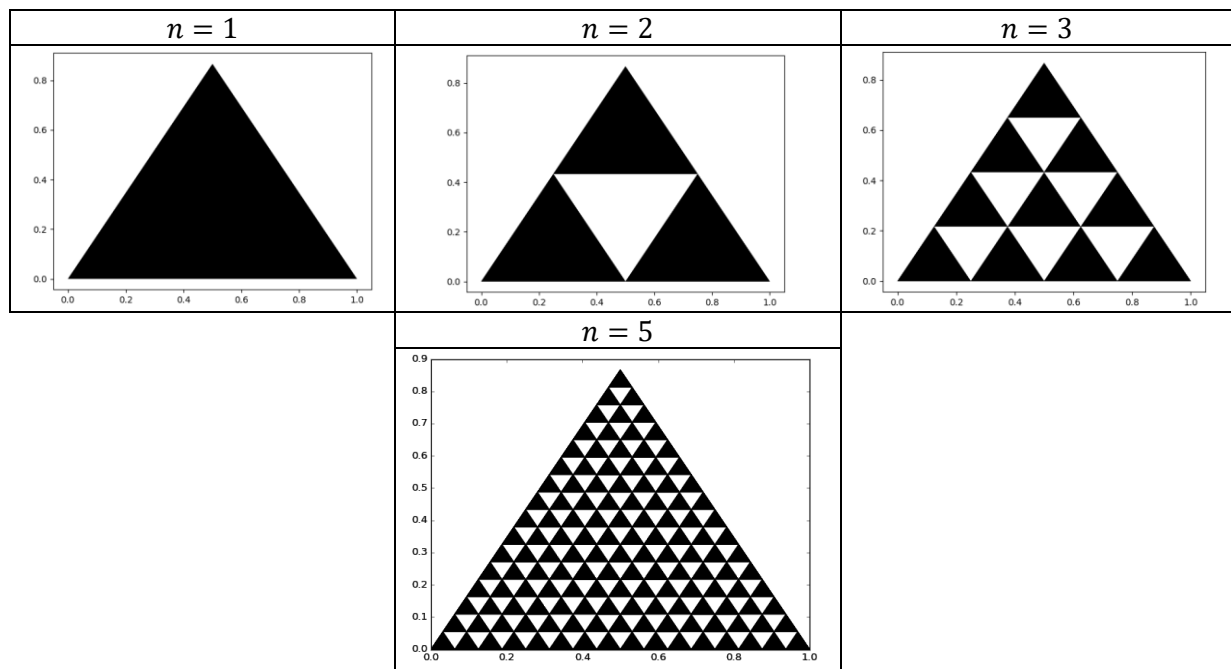
Remarques :

- Vous devrez modifier la fonction « Polygon » afin de prendre en compte un argument de couleur
- Vous remarquerez que l'ordre d'affichage des triangles est maintenant important puisqu'ils sont superposés... Un peu d'aide ? La fonction ne s'appelle qu'une fois à chaque étape !
- On ne trace qu'un triangle à chaque étape
- Vous pourrez dans un premier temps définir une couleur 'p' dans la fonction « Polygon » afin de faire varier la couleur aléatoirement à chaque triangle. Cela pourra vous permettre de voir que votre code fait plusieurs triangles les uns sur les autres, qui seraient invisibles s'ils étaient de la même couleur
- Vous pourrez utiliser un indice  $i$  valant 0 ou 1  $((i + 1))\%2$  pour choisir entre deux couleurs  $['k', 'w']$

Quand votre code fonctionne avec les couleurs aléatoires, amusez-vous à zoomer au centre, c'est comme une fractale ☺

Dernière mise à jour	Informatique	Claire GAUDY - Denis DEFAUCHY
12/01/2023	5 - Fonction récursives	TD 5-3 - Figures récursives

Voici enfin les images suivantes :



**Question 2:** En utilisant le code de la fonction `Triangles_Infinis` et en lui ajoutant 3 lignes, créer une fonction récursive `Triangles` réalisant ces images

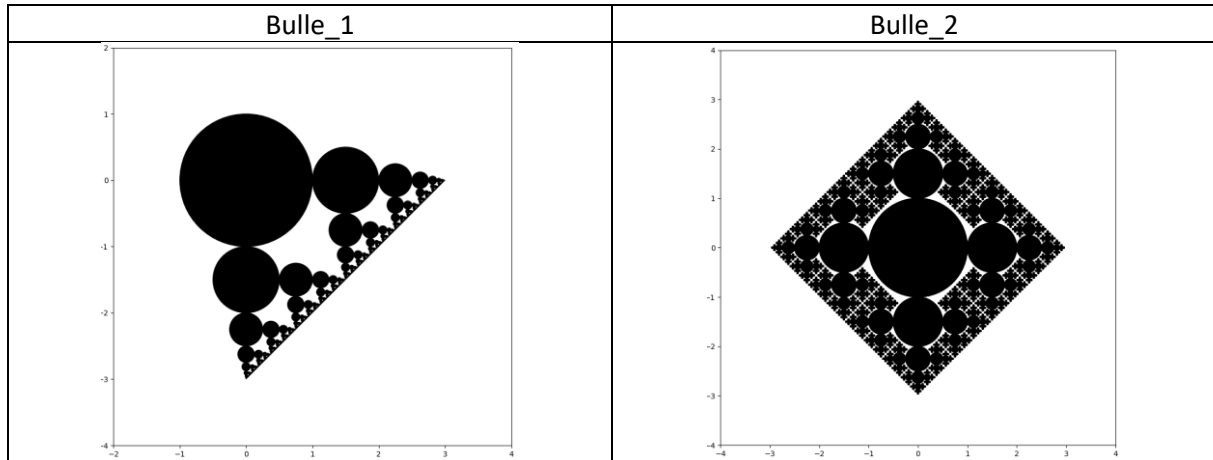
Remarque : '`p`' renvoie une couleur aléatoire parmi quelques couleurs de base, amusez-vous !



Dernière mise à jour	Informatique	Claire GAUDY - Denis DEFAUCHY
12/01/2023	5 - Fonction récursives	TD 5-3 - Figures récursives

## Exercice 5: Cercles récursifs

Soient les figures ci-dessous :



On souhaite les obtenir par récursivité.

Soit le code suivant :

```
import matplotlib.pyplot as plt
plt.close('all')

def Cercle(x,y,r):
    Circle = plt.Circle((x,y),r,color='k')
    plt.gca().add_artist(Circle)

def Affiche(x_min,x_max,y_min,y_max):
    plt.xlim(x_min,x_max)
    plt.ylim(y_min,y_max)
    plt.show()
    plt.gca().set_aspect('equal', adjustable='box')

def Bulles_1(n):
    def ...
    ...
    plt.figure()
    ...
    Affiche(-2,4,-4,2)

def Bulles_2(n):
    def ...
    ...
    plt.figure()
    ...
    Affiche(-4,4,-4,4)

n = 5
Bulles_1(n)
Bulles_2(n)
```

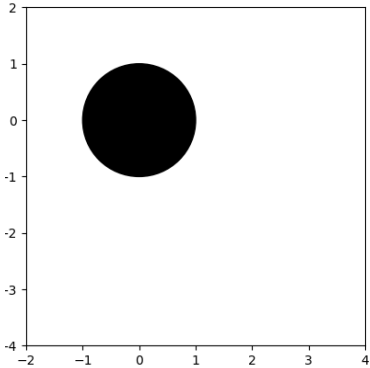
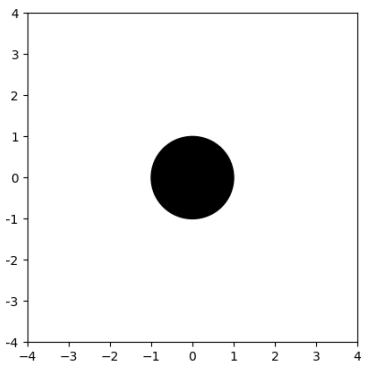
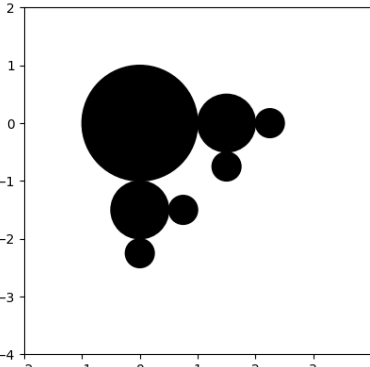
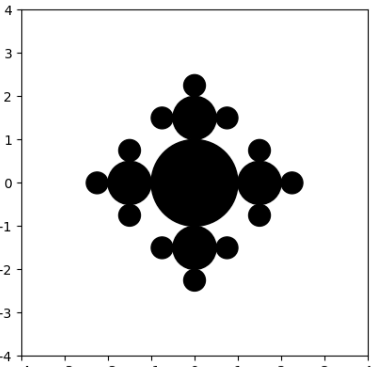
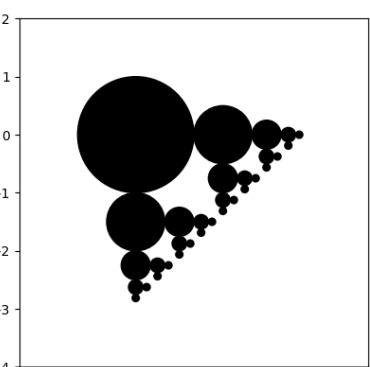
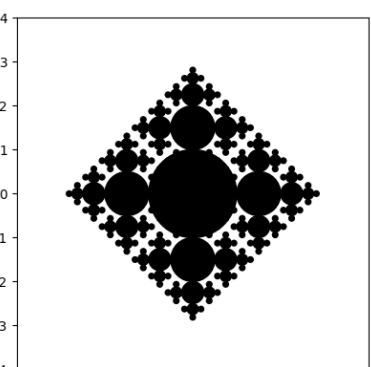
Dernière mise à jour	Informatique	Claire GAUDY - Denis DEFAUCHY
12/01/2023	5 - Fonction récursives	TD 5-3 - Figures récursives

Après avoir créé une figure vide avec « `plt.figure()` », les appels successifs de la fonction « `Cercle(x,y,r)` » ajoutent à la figure les cercles voulus sans les afficher.

Enfin, l'appel de la fonction « `Affiche(x_min,x_max,y_min,y_max)` » trace la figure dans l'intervalle demandé et remet le repère orthonormé avec equal.

**Question 1: Compléter la fonction Bulles\_1 afin de réaliser l'image associée de manière récursive**

**Question 2: Compléter la fonction Bulles\_2 afin de réaliser l'image associée de manière récursive**

N	Bulle_1	Bulle_2
1		
3		
5		

Remarque : pour Bulle\_2, l'image ci-dessus correspond à une méthode qui, à chaque nouvel appel, crée les 4 cercles associés.

**Question 3: Si vous avez fait cela, programmer à une méthode ne réalise le tracé que dans les 3 bonnes directions**

De l'aide : exploitez le fait que si  $\vec{u} = (u_x, u_y)$ ,  $\vec{n} = (\pm u_y, u_x)$  est orthogonal à  $\vec{u}$ .