**Notre Dame de La Merci**

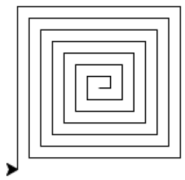
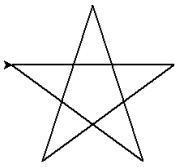
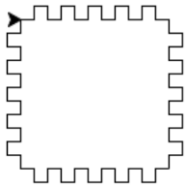
## Exercices sur Python Turtle

Commandes de base Python Turtle-Tortue

| | |
|---|---|
| up() | : lève le crayon |
| down() | : baisse le crayon |
| forward(n) | : avance de n |
| left(d) | : tourne vers la gauche de d degrés |
| right(d) | : tourne vers la droite de d degrés |
| goto(x,y) | : se déplace vers le point de coordonnées (x,y) |
| circle(r) | : dessine un cercle de rayon r |
| width(e) | : définit l'épaisseur du trait |
| speed(''texte'') | : définit la vitesse de la tortue |
| write(''texte'') | : écrit le texte |
| color(''couleur'') | : définit la couleur du trait |
| bgcolor(''couleur'') | : définit la couleur de fond |
| reset() | : efface tout |
| done() | : arrête le dessin |

**Exercice 1B.1 :**

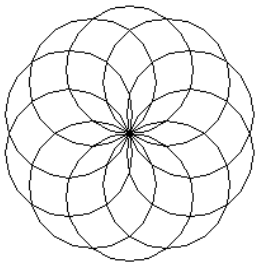
Représenter les schémas suivants :

a)  b)  c) 

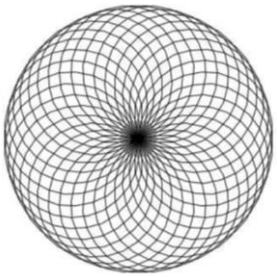
**Exercice 1B.2 :**

Représenter les schémas suivants :

Ils sont constitués de cercles de même rayon (80 dans notre cas), avec un décalage régulier entre deux cercles successifs.

a) 

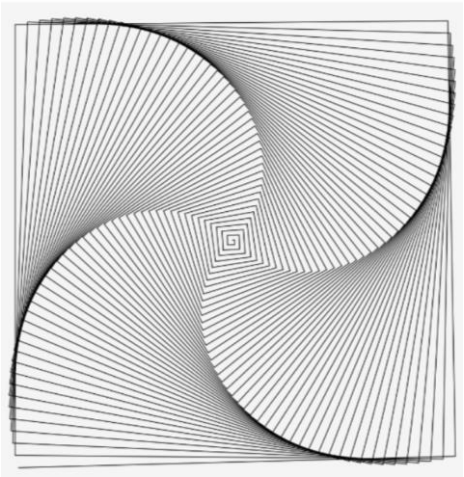
10 cercles

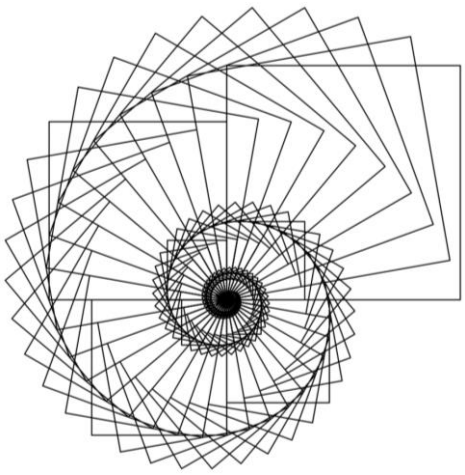
b) 

36 cercles

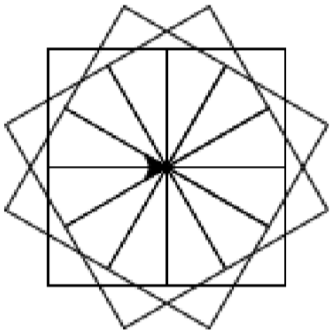
**Exercice 1B.3 :**

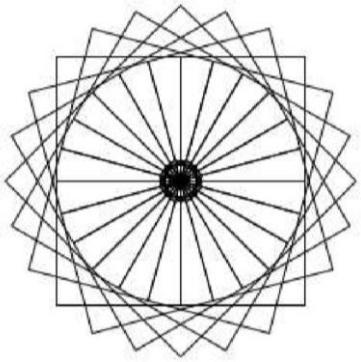
Représenter les schémas suivants :



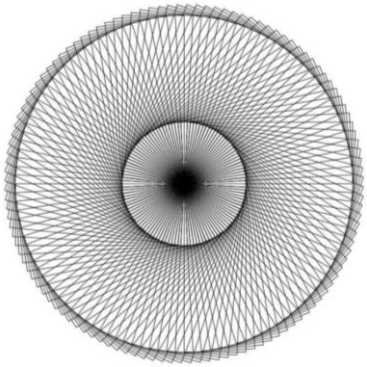


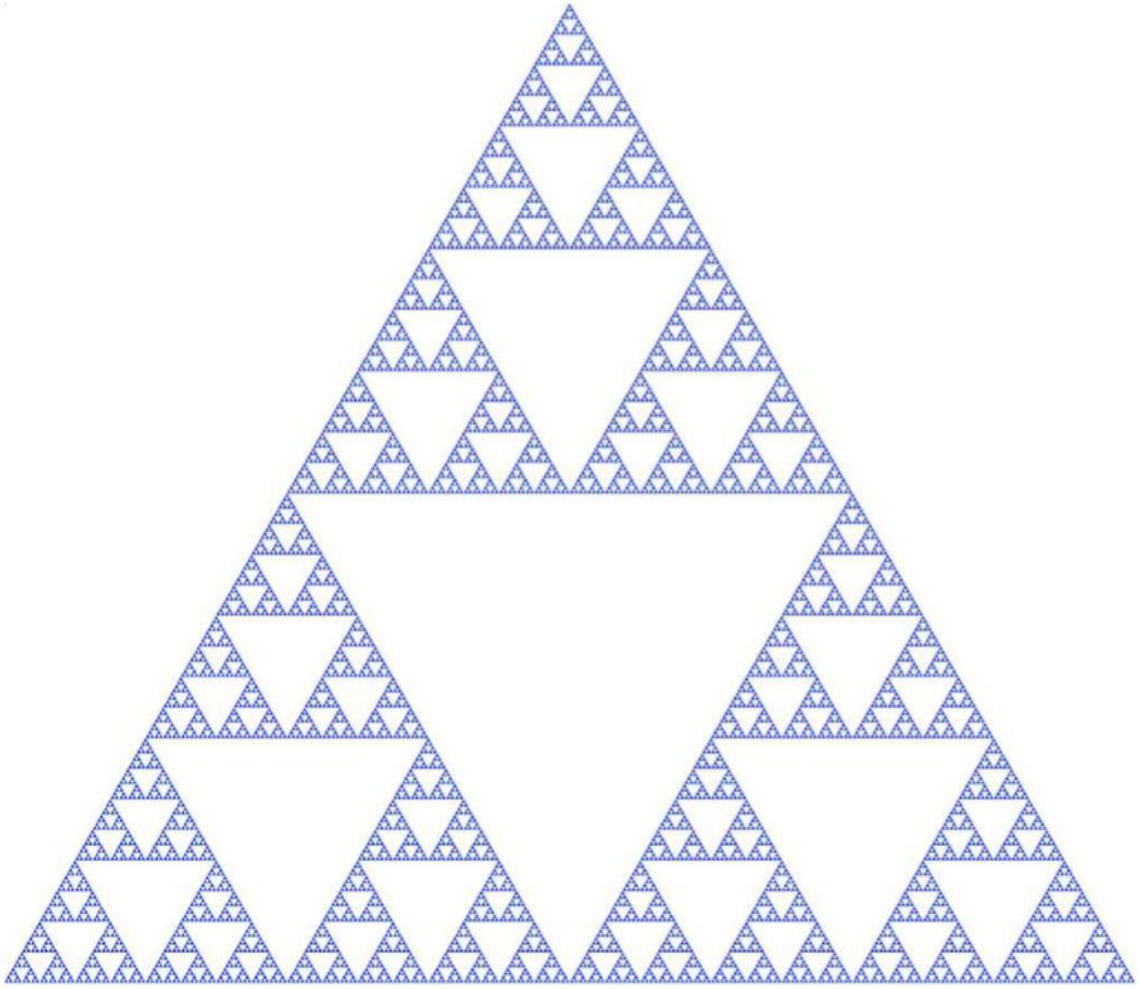
**Exercice 1B.4 :**
Représenter les schémas suivants :
rotations de carrés

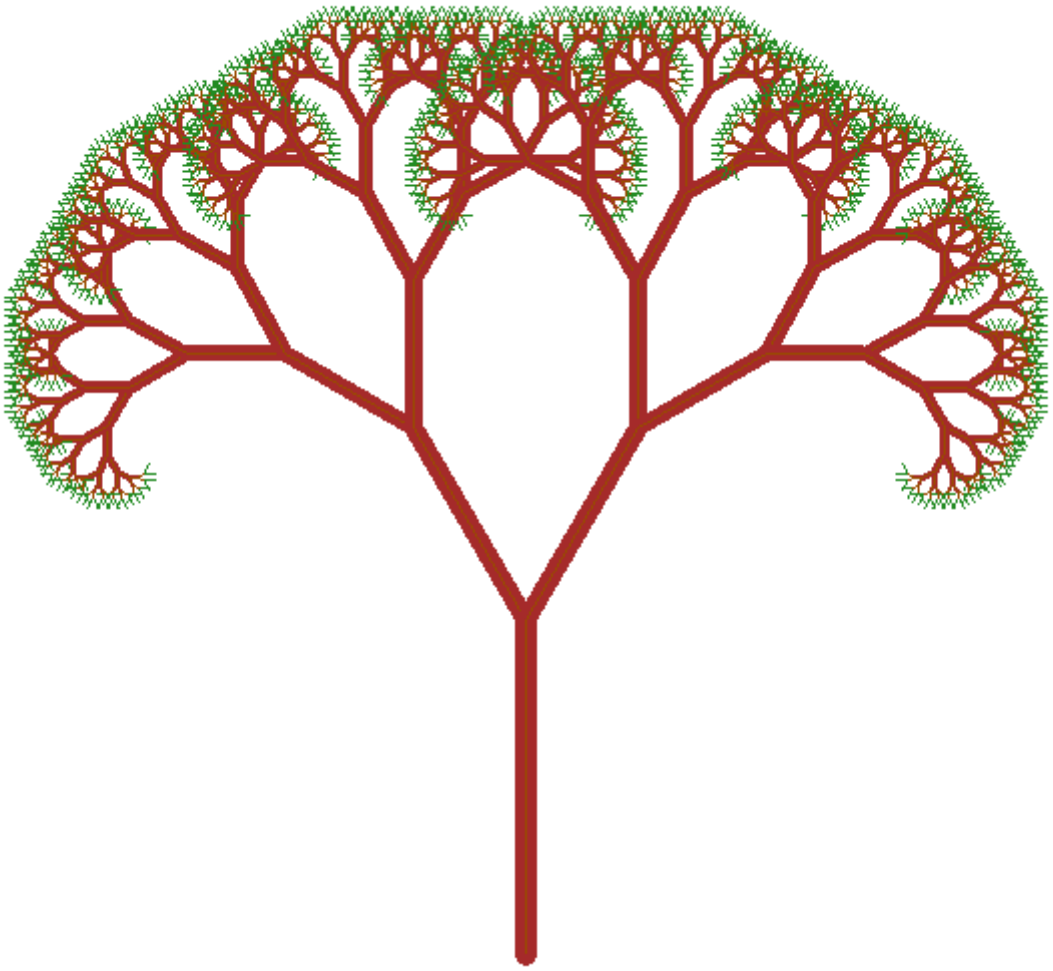




**Exercice 1B.5 :**
Représenter le schéma suivant :
rotation d'un rectangle autour de son centre







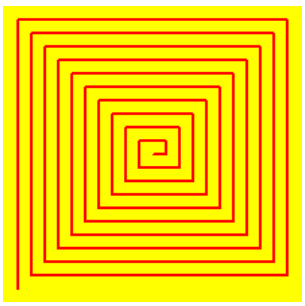
Arbre de Pythagore

**CORRIGE – Notre Dame de La Merci - Montpellier**

**Exercice 1B.1 a:**

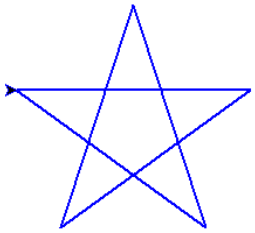
```
from turtle import *
n = int(input("Nombre  d'enroulements :"))
speed(10)
pensize(2)
pencolor("red")
bgcolor("yellow")
for i in range(1,2*n+1):
    forward(10*i)
    left(90)
    forward(10*i)
    left(90)
hideturtle()
mainloop()
```

Pour n = 10 :



**Exercice 1B.1 b:** Etoile à n branches

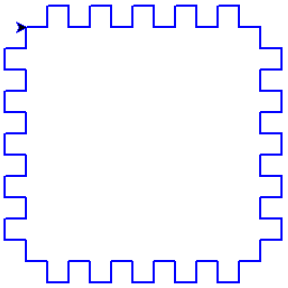
```
from  turtle  import *
n = int(input("Nombre  de branches :"))
pensize(2)
pencolor("blue")
for i in range (n):
    forward(200)
    right(180-180/n)
mainloop ()
```



*Ce programme ne marche pas pour un nombre pair de branches (!)*

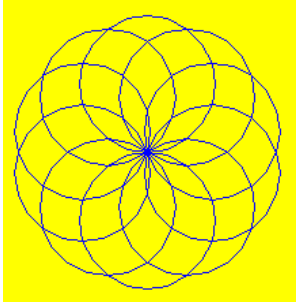
**Exercice 1B.1 c:** carré avec cannelures

```
from  turtle  import *
pensize(2)
pencolor("blue")
for i in range (4):
    for j in range (5):
        forward(20)
        left(90)
        forward(20)
        right(90)
        forward(20)
        right(90)
        forward(20)
        left(90)
    forward(20)
    right(90)
mainloop ()
```

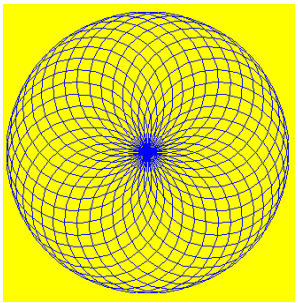


**Exercice 1B.2 :** cercles imbriqués

```
from turtle import *
n = int(input("Nombre  de cercles :"))
speed(10)
pencolor("blue")
bgcolor("yellow")
for i in range(n):
    circle(50)
    left(360/n)
hideturtle()
mainloop()
```



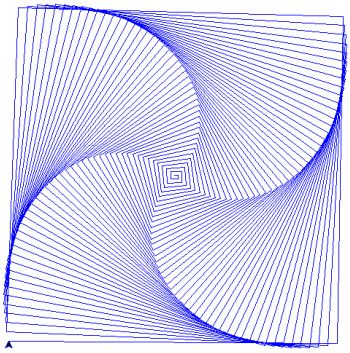
10 cercles



36 cercles

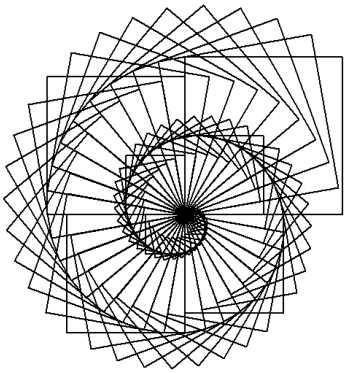
**Exercice 1B.3 a:**

```
from  turtle  import *
pencolor("blue")
speed(10)
for i in range (1,91):
    for j in range(2):
        forward(5*i)
        right(89.5)
mainloop ()
```



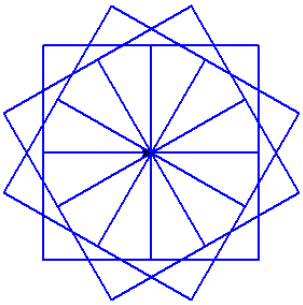
**Exercice 1B.3 b:**

```
from turtle import *
n=73
speed(20)
pensize(2)
pencolor("black")
for i in range(n):
    for j in range(4):
        forward(3*i)
        left(90)
    right(10)
```



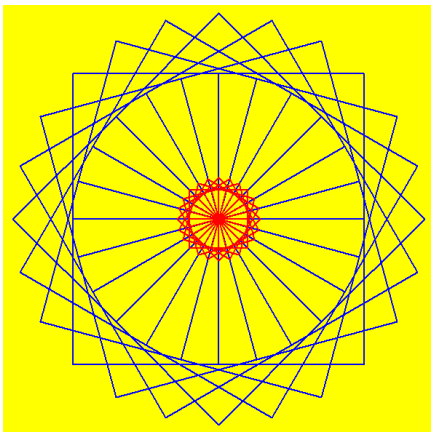
**Exercice 1B.4 a:**

```
from turtle import *
speed(10)
pensize(2)
pencolor("blue")
for i in range(12):
    for j in range(4):
        forward(100)
        left(90)
    right(30)
```



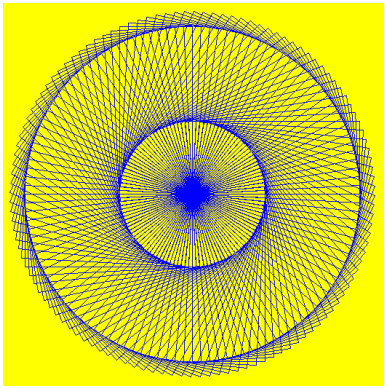
**Exercice 1B.4 b:**

```
from turtle import *
speed(20)
pencolor("blue")
bgcolor("yellow")
for i in range(24):
    pensize(2)
    for j in range(4):
        forward(200)
        left(90)
    right(15)
for i in range(24):
    pensize(1)
    for j in range(4):
        pencolor("red")
        forward(40)
        left(90)
    right(15)
hideturtle()
```



**Exercice 1B.5 :**

```
from turtle import *
speed(20)
pensize(1)
pencolor("blue")
bgcolor("yellow")
for i in range(120):
    for j in range(2):
        forward(230)
        left(90)
        forward(100)
        left(90)
    right(3)
hideturtle()
```



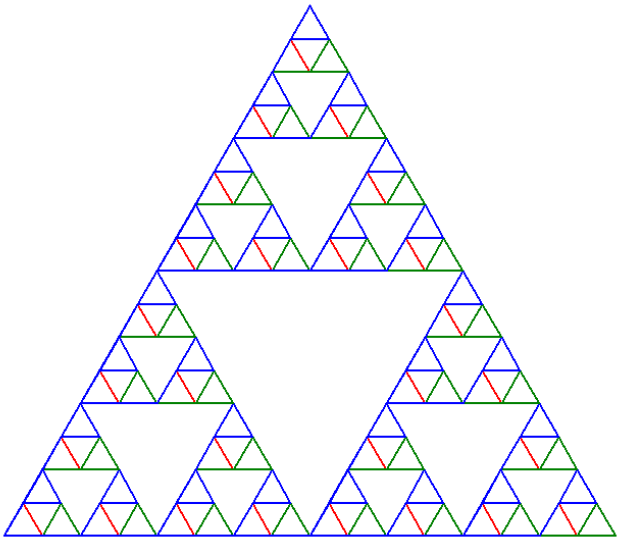
**Fractale :**

Partir du grand triangle et le subdiviser successivement, d'abord en 4, puis en 13,…

```
from  turtle  import *
n = int(input("Nombre  de divisions :"))
pensize(2)
pencolor("blue")
up()
setposition(-300,-300)
down()
for i in range (3):
   forward(720)
   left(120)
mainloop ()
```

```
def sierpinski(n,longueur):
    speed(0)#speed du traceur au maximum
    pencolor("black")#couleur du stylo
    shape("turtle")#gadjet
    ht()#masque la flèche qui trace
    if n==0:
      for i in range(0,3):
            forward(longueur)
            left(120)
    if n>0:
      sierpinski(n-1,longueur/2)
      forward(longueur/2)
      sierpinski(n-1,longueur/2)
      backward(longueur/2)
      left(60)
      forward(longueur/2)
      right(60)
      sierpinski(n-1,longueur/2)
      left(60)
      backward(longueur/2)
      right(60)
decale_vers_la_gauche(300)
decale_vers_le_haut(-200)
sierpinski(4,600)
```

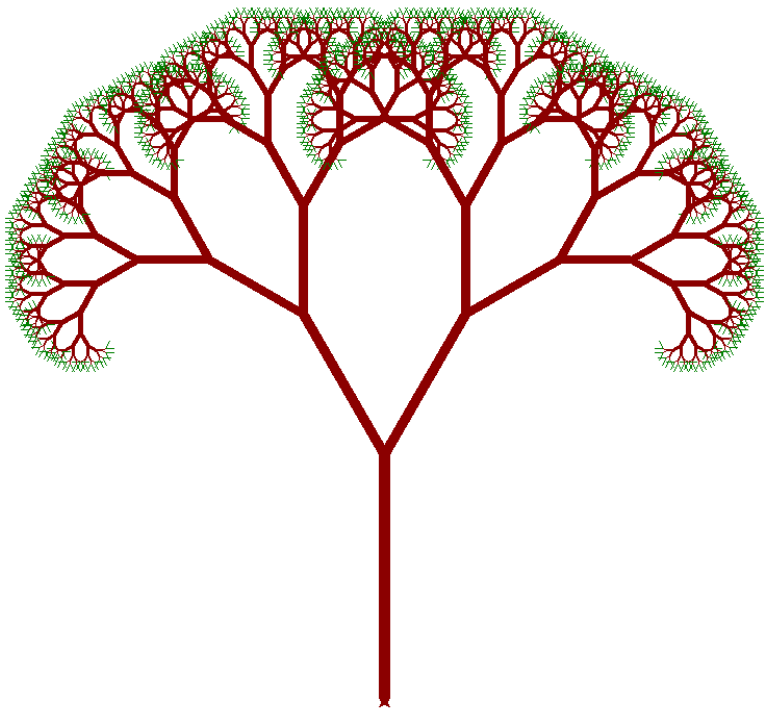
```
import turtle

T = turtle.Turtle() ; T.speed(0)
T.hideturtle()

def Sier (n,L):
    if n == 0:
        for i in range (0,3):
            T.fd(L)
            T.left(120)
    if n > 0:
        T.pencolor(co[0])
        Sier(n-1,L/2)
        T.fd(L/2)
        T.pencolor(co[1])
        Sier(n-1,L/2)
        T.bk(L/2)
        T.left(60)
        T.fd(L/2)
        T.right(60)
        T.pencolor(co[3])
        Sier(n-1,L/2)
        T.left(60)
        T.bk(L/2)
        T.right(60)

T.penup(); T.goto(-400,-300)
T.pendown() ; T.width(2)
co = ["red", "green", "purple", "blue"]
T.pencolor(co[0])
Sier(4,600)
```



## CORRIGE DU NET

## Arbre de Pythagore :

```
from turtle import *
speed (100)
angle =30
color("darkred")
def arbre(n, longueur ):
    if n == 0:
        color("green")
        forward(longueur)
        backward(longueur)
        color("darkred")
    else:
        width(n)
        forward(longueur / 3)
        left(angle)
        arbre(n - 1, longueur / 3 * 2)
        right (2* angle)
        arbre(n - 1, longueur / 3 * 2)
        left(angle)
        backward(longueur / 3)
```



```
hideturtle ()
up()
right(90)
forward(300)
left (180)
down()
arbre (11, 700)
showturtle ()
mainloop ()
```

→ $2^n$ feuilles pour l'arbre