

```

# ===== #
# ===== #
# IPT MATHS SPE #
# Programme de tracé de la spirale de Fibonacci #
# Novembre 2017 #
# ===== #
# ===== #

## Importations
from numpy import linspace
from math import pi, sin, cos
import matplotlib.pyplot as plt

## Définitions des fonctions
def QuartCercle(xc,yc,teta0,R,N=101,sens='t'):
    if sens == 't':
        teta1 = teta0 + pi/2
    else:
        teta1 = teta0 - pi/2
    T = linspace(teta0,teta1,N)
    X, Y = [], []
    for i in range(N):
        X.append(xc + R * cos(T[i]))
        Y.append(yc + R * sin(T[i]))
    plt.plot(X,Y,color="red",linewidth=2)

def SpiraleFibo(n,xc_current,yc_current,teta=0,R=0,R2=1,sens='t'):
    if n == 0:
        # Cas de base. On trace !
        plt.title("Spirale de Fibonacci",size=30)
        plt.grid()
        plt.axis('equal')
        plt.show()
    else:
        # Quart de cercle courant
        QuartCercle(xc_current,yc_current,teta,R2,101,sens)
        if sens == 't':
            eps = 1.
        else:
            eps = -1.
        teta1 = teta + eps * pi/2
        x_last = xc_current - eps * R2 * sin(teta)
        y_last = yc_current + eps * R2 * cos(teta)
        # Rayon du prochain quart de cercle
        new_R = R + R2
        # Coordonnées du centre du prochain quart de cercle
        xc_new = x_last - new_R * cos(teta1)
        yc_new = y_last - new_R * sin(teta1)
        # Appel récursif
        SpiraleFibo(n-1,xc_new,yc_new,teta1,R2,new_R)

## Pour tester...
plt.clf()
SpiraleFibo(11,0,1,-pi/2)

```