

```

# -*- coding: utf-8 -*-
"""
Suite de Fibonacci.

Méthode itérative.
Dépendance du temps de calcul avec le rang n.
"""

# Importations des bibliothèques
from time import perf_counter
import matplotlib.pyplot as plt

# Fonction itérative pour le calcul de F(n).
# Comme on s'intéresse à de grands rangs, les cas n=0 et n=1 ne sont pas pris
# en compte.
def fibo_iter(a,b,n):
    for i in range(n):
        a,b = b,a+b
    return(b)

# PROGRAMME PRINCIPAL
# =====

# Saisie des paramètres par l'utilisateur :
# n1 : rang initial
# n2 : rang final
# dn : pas
n1 = int(input('Veuillez saisir le rang initial : '))
n2 = int(input('Veuillez saisir le rang final : '))
dn = int(input('Veuillez saisir le pas : '))

# Création des listes des pour la génération du graphique
x = [i for i in range(n1,n2+dn,dn)]
y = []

for n in range(n1,n2 + dn,dn):
    t_init = time.perf_counter()
    res = fibo_iter(0,1,n)
    t_final = time.perf_counter()
    y.append(t_final - t_init)

# On efface la figure courant
plt.clf()

# Génération des points du graphique
plt.plot(x,y,'.',color='red')

# Génération des principaux éléments du graphique (titre et légendes sur les axes)
titre = 'Temps de calcul du terme de rang n\nde la suite de Fibonacci\n(méthode
itérative).'
plt.title(titre)
plt.xlabel('n')
plt.ylabel('Temps de calcul (secondes).')

# Affichage du graphique
plt.show()

# FIN DU PROGRAMME PRINCIPAL
# =====

```