

```

import numpy as np
from numpy import sqrt
import networkx as nx
import matplotlib.pyplot as plt

def premier(atester):
    k = 2
    if atester in [0, 1]: return False
    if atester in [2, 3, 5, 7]: return True
    while True:
        if k * k > atester: return True
        else:
            if atester % k == 0: return False
            else: k = k + 1

for n in range(98,100,2):
    matrice = np.zeros((n-1,n-1),dtype='bool')
    print('\n',n,':::')
    for d in range(n//2+1):
        if d == 0 or premier(d+1):
            for k in range(((n-2*d-2)//(d+1))+1):
                if n-2*d-(1+k*(d+1)) >= 0:
                    i = 1+k*(d+1)-1
                    j = n-2*(d+1)-i
                    matrice[i, j] = True
    noncommut = np.zeros((n-1,n-1),dtype='bool')
    for x in range(len(matrice)):
        for y in range(len(matrice)):
            if matrice[x,y] and not matrice[y,x]:
                noncommut[x,y] = True
    for x in range(len(noncommut)):
        for y in range(len(noncommut)):
            if noncommut[x,y]:
                print('(',x,',',y,')', ' ',end='')
    print('')
    G = nx.from_numpy_array(noncommut)
    #nx.draw_circular(G,node_color='red',edge_color='#909090',node_size=100)
    plt.show()
    for x in range(n-1):
        if G.degree[x] == 0 and x%2 != 0 and x > sqrt(n):
            print(x, ' est de degre 0.',end='')
            if premier(x) and premier(n-x):
                print(' dg.')
        else:
            print(' rate.')

```