

```

"""
Created on Wed Mar  4 18:27:15 2020
@author: campioni - Cap 11
file: Kauffmann nota 8"""

'''importa vari moduli
'''

import networkx as nx
import numpy as np

G=nx.Graph()
print()
'''
=====
PARAMETRI DI INGRESSO
=====
'''

N=int(input('N = '))

k=int(input('k = '))
print()
f=k+1

def ada(N): #Calcola la media di w1 , w2 , w3
    w=0
    for h in range(N):
        w=w+np.random.rand()/(N)
    return w

def LISTA_ADA(N): #Costrisce una lista casuale di 2**(k+1) valori
    A=[]
    for i in range (2**f):
        a=round(ada(N),5)
        A.append(a)
    return A

P=[]          # Inizializza i parametri
picco=[]
Z=[]
nodo=[]
c=0

A=LISTA_ADA(N)

G = nx.random_regular_graph(k, 2**f, seed=None) # definisce il tipo di grafo
nx.draw_circular(G,node_color='red',
                 node_size=1000,
                 with_labels=True)

for i in range (2**f): #P è la lista dei vicini
    P.append([n for n in G.neighbors(i)])
    P[i].insert(0,i)

for h in range(2**f): #In Z sono messi i vicini presi da A
    for i in P[h]:     #e identificati da P
        Z.append(A[i])

#Si cercano i punti ottimali intermedi e si pongono in <<picco>>

for h in range(2**f):

```

```

    if Z[f*h]==np.max(Z[f*h:f*h+f]):
        picco.append(Z[f*h])
        nodo.append(h)
        c+=1

'''
=====
Si stampano i risultati
=====
'''
if k==0:
    print ('Unico picco ', '%5f' % np.max(Z))
else:
    print('I picchi relativi sono: ', c, ', specificatamente:')
    print()
    for i in range(c):
        print('Nodo ', nodo[i], ' --- valore ', '%5f' % picco[i])

    print()
    print('=====')
    print()
    print('Il picco massimo è: al nodo n° ', picco.index(np.max(picco)), ', ',
valore ' '%5f' % np.max(Z))

```