```
# -*- coding: utf-8 -*-
"""
@author: campioni
"""
import networkx as nx
from numpy.random import random

def flip(p):     #da' vero per una probabilita' p, falso per 1-p
    return random()<p

def random_pairs(nodes, p):        #aggiunge archi al grafo con probabilità p
    for i, u in enumerate(nodes):
        for j, v in enumerate(nodes):
            if i<j and flip(p):
               yield u, v

def make_random_graph(n, p):    #genera un grafo casuale ER
    G = nx.Graph()
    nodes = range(n)
    G.add_nodes_from(nodes)
    G.add_edges_from(random_pairs(nodes, p))
    return G

n=int(input('numero di nodi = '))
p=float(input("probabilita' = "))

random_graph = make_random_graph(n,p)
nx.draw_circular(random_graph,
                 node_color=COLORS[2],
                 node_size=1000,
                 with_labels=True)
print()
print('numero di archi = ',len(random_graph.edges()))
```