ScriptScriptScriptScript

In [72]:

%matplotlib inline

import networkx as nx

import matplotlib.pyplot as plt

import numpy as np

F=nx.powerlaw\_cluster\_graph (9 , 1, 0.4)

G=nx.barabasi\_albert\_graph (100 , 2)

H=nx.grid\_2d\_graph (10 , 10)

I=nx.complete\_graph (10)

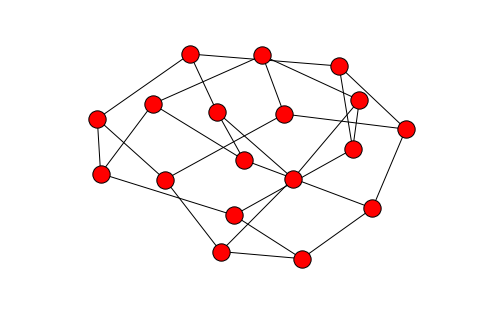
J=nx.cycle\_graph (10)

K=nx.erdos\_renyi\_graph (100 , 0.01)

L=nx.pappus\_graph ()

nx.draw(L)

plt.show()



In [73]:

print("prediction","[0,0,0,18]")

deg = nx.degree\_histogram(L)

print(deg)

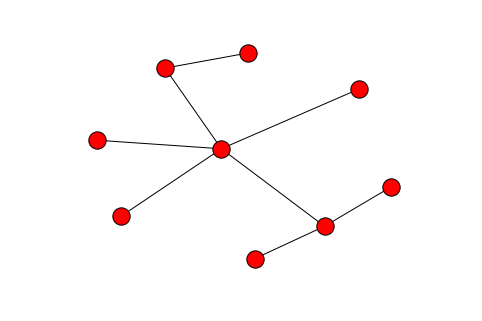
prediction [0,0,0,18]

[0, 0, 0, 18]

In [74]:

nx.draw(F)

plt.show()



In [76]:

print("hard to make prediction")

deg = nx.degree\_histogram(F)

print(deg)

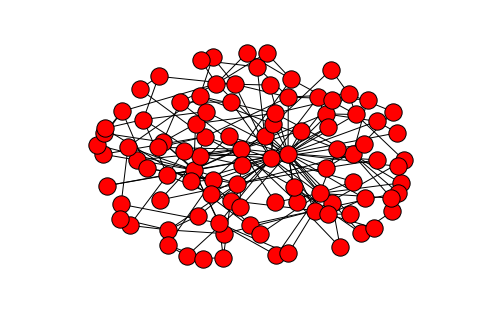
hard to make prediction

[0, 6, 1, 1, 0, 1]

In [77]:

nx.draw(G)

plt.show()



In [78]:

print("hard to make prediction")

deg = nx.degree\_histogram(G)

print(deg)

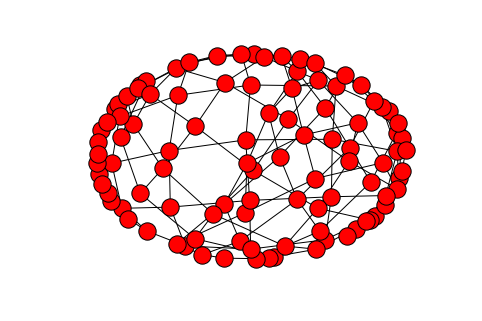
hard to make prediction

[0, 0, 49, 13, 16, 7, 4, 4, 2, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

In [79]:

nx.draw(H)

plt.show()



In [80]:

print("prediction", "[0, 0, 4, 32, 64]")

deg = nx.degree\_histogram(H)

print(deg)

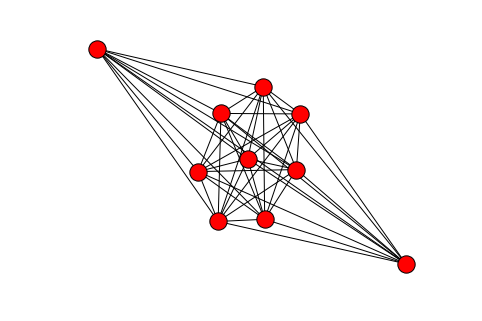
prediction [0, 0, 4, 32, 64]

[0, 0, 4, 32, 64]

In [81]:

nx.draw(I)

plt.show()



In [82]:

print("prediction", "[0, 0, 0, 0, 0, 0, 0, 0, 0, 10]")

deg = nx.degree\_histogram(I)

print(deg)

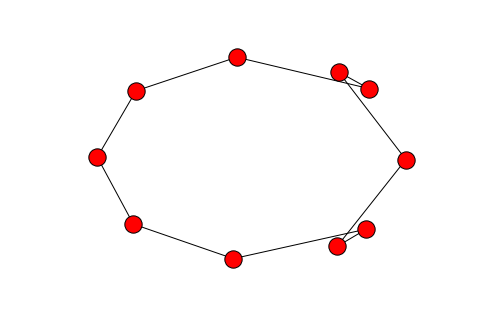
prediction [0, 0, 0, 0, 0, 0, 0, 0, 0, 10]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 10]

In [83]:

nx.draw(J)

plt.show()



In [84]:

print("prediction","[0,0,10]")

deg = nx.degree\_histogram(J)

print(deg)

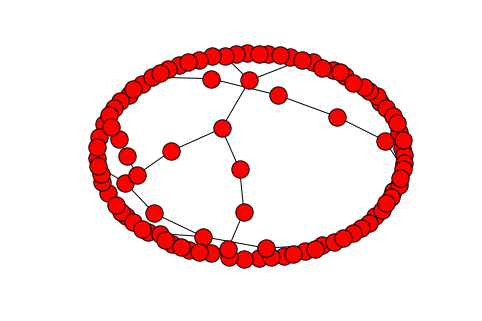
prediction [0,0,10]

[0, 0, 10]

In [85]:

nx.draw(K)

plt.show()



In [86]:

print("hard to make prediction")

deg = nx.degree\_histogram(K)

print(deg)

hard to make prediction

[38, 33, 19, 7, 3]

In [ ]: