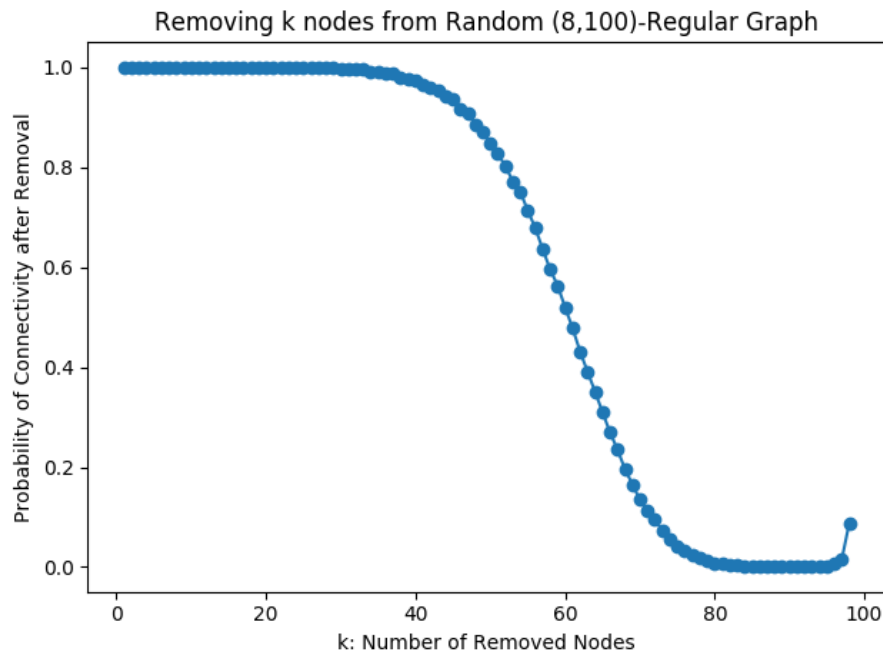


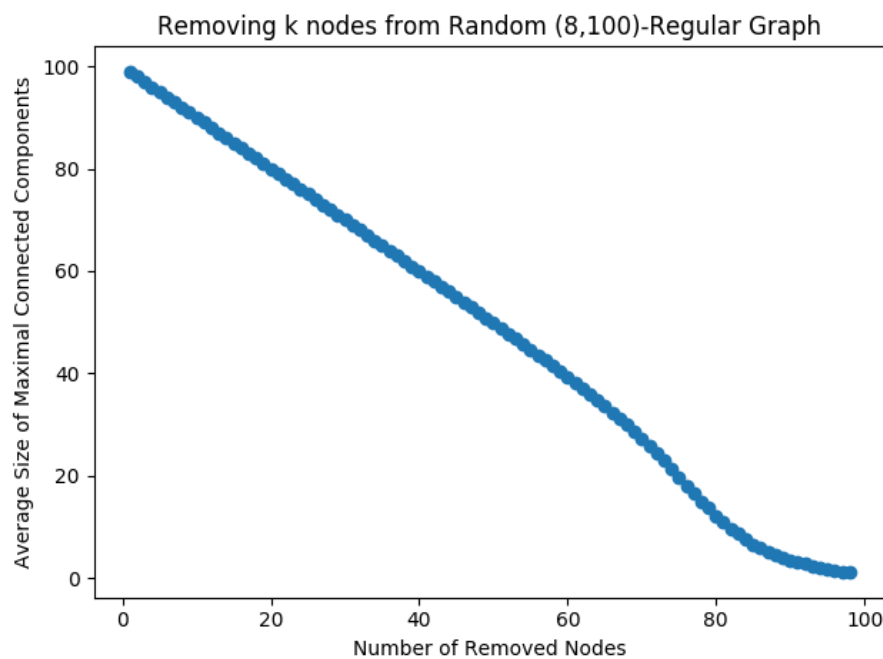
Task 01: Deleting K Nodes from (8,100) regular graph

Randomly generated 8-regular graph with 100 nodes

- Randomly delete k nodes from graph, the probability that the remaining graph is still connected is shown in graph below. (for each k , the experiment is repeated 8000 times)



- Randomly delete k nodes from graph, the average size of the maximal connected components in remaining graph is shown in graph below. (for each k , the experiment is repeated 8000 times)



```

import networkx as nx
import matplotlib.pyplot as plt
import random

k = 8 # the degree of the network is 8
n = 100 # the network has 100 nodes
m = 8000 # the repeat run

avergae_com = []
conn_prob = []
for i in range (1, n - 1):
    num = 0
    com_n = 0
    for j in range(m):
        rg = nx.random_graphs.random_regular_graph(k, n)
        del_nodes = random.sample(range(0, n), i)
        for node in del_nodes:
            rg.remove_node(node)
        largest_com = max(nx.connected_components(rg), key=len)
        len_com = len(largest_com)
        com_n += len_com
        if len_com == n - i:
            num += 1
    avergae_com.append(com_n / m)
    conn_prob.append(num / m)

plt.plot(range (1, n - 1), conn_prob, 'o-')
plt.xlabel('k: Number of Removed Nodes')
plt.ylabel('Probability of Connectivity after Removal')
plt.title('Removing k nodes from Random (8,100)-Regular Graph')
plt.show()

plt.plot(range (1, n - 1), avergae_com, 'o-')
plt.xlabel('Number of Removed Nodes')
plt.ylabel('Average Size of Maximal Connected Components')
plt.title('Removing k nodes from Random (8,100)-Regular Graph')
plt.show()

```