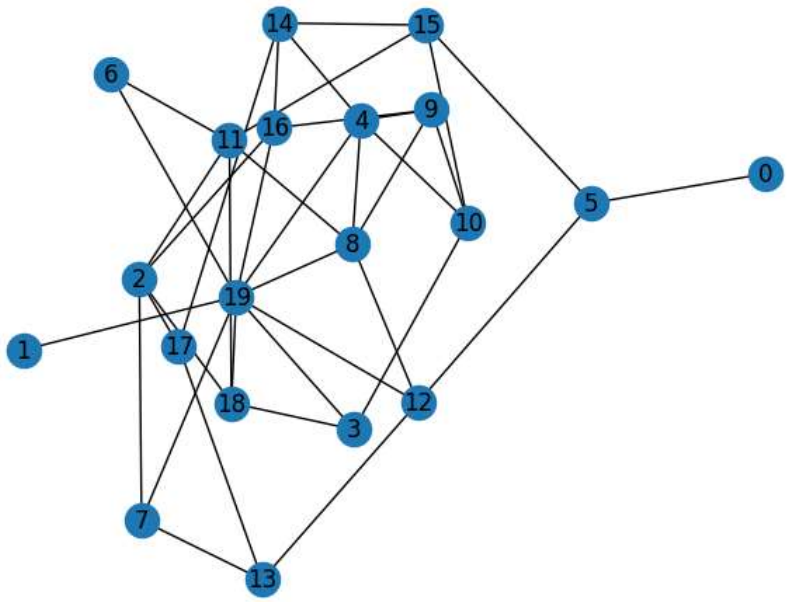
```
import networkx as nx
import matplotlib.pyplot as plt
import numpy as np


# Generate random social network
def generate_social_network(num_nodes, avg_degree):
    return nx.erdos_renyi_graph(num_nodes, avg_degree / num_nodes)


# Compute centrality measures
def compute_centralities(graph):
    degree_cent = nx.degree_centrality(graph)
    closeness_cent = nx.closeness_centrality(graph)
    betweenness_cent = nx.betweenness_centrality(graph)
    return degree_cent, closeness_cent, betweenness_cent


# Find most influential nodes based on centrality measures
def most_influential_nodes(centrality_measure, top_n=5):
    sorted_nodes = sorted(centrality_measure.items(), key=lambda x: x[1], reverse=True)[:top_n]
    return [node[0] for node in sorted_nodes]


# Visualize network
def visualize_network(graph):
    pos = nx.spring_layout(graph)
    nx.draw(graph, pos, with_labels=True)
    plt.show()


# Parameters
num_nodes = 20
avg_degree = 4

# Generate and visualize random social network
social_network = generate_social_network(num_nodes, avg_degree)
visualize_network(social_network)
```




```
# Compute centrality measures
degree_centrality, closeness_centrality, betweenness_centrality = compute_centralities(social_network)
```

```
# Find most influential nodes
top_degree_centrality = most_influential_nodes(degree_centrality)
top_closeness_centrality = most_influential_nodes(closeness_centrality)
top_betweenness_centrality = most_influential_nodes(betweenness_centrality)


# Print results
print("Degree Centrality:")
print(degree_centrality)
print("Most influential nodes based on Degree Centrality:", top_degree_centrality)
print("\nCloseness Centrality:")
print(closeness_centrality)
print("Most influential nodes based on Closeness Centrality:", top_closeness_centrality)
print("\nBetweenness Centrality:")
print(betweenness_centrality)
print("Most influential nodes based on Betweenness Centrality:", top_betweenness_centrality)
```

```
Degree Centrality:
{0: 0.05263157894736842, 1: 0.05263157894736842, 2: 0.2631578947368421, 3: 0.15789473684210525, 4: 0.2631578947368421, 5: 0.15789
Most influential nodes based on Degree Centrality: [19, 2, 4, 8, 11]

Closeness Centrality:
{0: 0.30158730158730157, 1: 0.38, 2: 0.475, 3: 0.4634146341463415, 4: 0.5, 5: 0.4222222222222222, 6: 0.4318181818181818, 7: 0.452
Most influential nodes based on Closeness Centrality: [19, 8, 11, 12, 4]

Betweenness Centrality:
{0: 0.0, 1: 0.0, 2: 0.08074352548036758, 3: 0.023424301494476933, 4: 0.06489371577090874, 5: 0.11848602988953867, 6: 0.0053606237
Most influential nodes based on Betweenness Centrality: [19, 12, 5, 15, 11]
```