Question 1: Shortest Path in a Road Network (Dijkstra's Algorithm)
A logistics company called Home Logistics wants to determine the most efficient route between two cities in a given road network. The network is represented as a graph where cities are nodes and roads are edges with weights corresponding to the travel distance (in kilometers).
Given the following graph representation of a road network, write a Python program using Dijkstra's Algorithm to find the shortest path from City A to City F.

Graph Data (as adjacency list):

```
roads = {
    'A': {'B': 4, 'C': 2},
    'B': {'A': 4, 'C': 5, 'D': 10},
    'C': {'A': 2, 'B': 5, 'D': 3, 'E': 8},
    'D': {'B': 10, 'C': 3, 'E': 6, 'F': 2},
    'E': {'C': 8, 'D': 6, 'F': 4},
    'F': {'D': 2, 'E': 4}
}
```

Question 2: Influence Analysis in a Social Network (PageRank Algorithm)
A social media platform wants to identify the most influential users based on follower relationships. The network is represented as a directed graph, where each user is a node, and an edge from user A to user B means that A follows B. Given the following directed graph of follower relationships, implement a Python program using the PageRank algorithm to rank users by influence.

Graph Representation:

```
followers = {
    'Alice': ['Bob', 'Charlie'],
    'Bob': ['Charlie', 'David'],
    'Charlie': ['David'],
    'David': ['Alice'],
    'Eve': ['Alice', 'Charlie']
}
```

Compute the PageRank scores and determine the most influential user.

Question 3: Maximum Flow in a Water Distribution System (Ford-Fulkerson Algorithm)
A city's water supply system consists of reservoirs, pipelines, and distribution points. The system is represented as a directed graph, where nodes represent junctions (reservoirs or city areas) and edges represent water pipelines with capacity limits. Given the following network, where the source is S (reservoir) and the sink is T (city distribution center), use the Ford-Fulkerson algorithm to determine the maximum amount of water that can be transported to the city.

Graph Representation (with capacities):

```
water_network = {
```

```
    'S': {'A': 16, 'B': 13},
    'A': {'B': 10, 'C': 12},
    'B': {'D': 14},
    'C': {'B': 9, 'T': 20},
    'D': {'C': 7, 'T': 4},
    'T': {}
}
```

Write a Python program to compute the maximum flow from **S** to **T**.