D. Case Study 4: Basic Graph Algorithms - Social Network Connections

6. Write a Python program to find mutual friends using BFS or DFS.

```
from collections import deque, defaultdict


def bfs(graph, start):
    visited = set()
    queue = deque([start])
    friends = set()


    while queue:
        node = queue.popleft()
        if node not in visited:
            visited.add(node)
            friends.update(graph[node])
            for neighbor in graph[node]:
                if neighbor not in visited:
                    queue.append(neighbor)
    return friends


def find_mutual_friends(graph, user1, user2):
    friends_of_user1 = bfs(graph, user1)
    friends_of_user2 = bfs(graph, user2)
    mutual_friends = friends_of_user1.intersection(friends_of_user2)
```

```
    return mutual_friends


# Example graph represented as an adjacency list
social_network = {
    'A': {'B', 'C', 'D'},
    'B': {'A', 'C', 'E'},
    'C': {'A', 'B', 'F'},
    'D': {'A'},
    'E': {'B'},
    'F': {'C'}
}


user1 = 'A'
user2 = 'C'
mutual_friends = find_mutual_friends(social_network, user1, user2)
print(f"Mutual friends of {user1} and {user2}: {mutual_friends}")
```