import matplotlib.pyplot as plt

import networkx as nx

def dfs(visited, graph, node):

    if node not in visited:

        print(node, end=" ")

        visited.add(node)

        for neighbour in graph[node]:

            dfs(visited, graph, neighbour)

def visualize\_graph(graph):

    G = nx.DiGraph()

    for node in graph:

        for neighbor in graph[node]:

            G.add\_edge(node, neighbor)

    pos = nx.spring\_layout(G)

    nx.draw(G, pos, with\_labels=True, node\_color="lightgreen",

            node\_size=1200, edge\_color="gray", arrows=True)

    plt.title("DFS Graph Visualization")

    plt.show()

def main():

    graph = {}

    visited = set()

    n = int(input("Enter number of nodes: "))

    for i in range(1, n + 1):

        edges = int(input(f"Enter number of edges for node {i}: "))

        graph[i] = []

        for j in range(1, edges + 1):

            node = int(input(f"Enter edge {j} for node {i}: "))

            graph[i].append(node)

    start\_node = int(input("Enter starting node for DFS: "))

    print("DFS Traversal:")

    dfs(visited, graph, start\_node)

    print()

    visualize\_graph(graph)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

output:

Enter number of nodes: 5

Enter number of edges for node 1: 3

Enter edge 1 for node 1: 2

Enter edge 2 for node 1: 3

Enter edge 3 for node 1: 4

Enter number of edges for node 2: 1

Enter edge 1 for node 2: 5

Enter number of edges for node 3: 2

Enter edge 1 for node 3: 4

Enter edge 2 for node 3: 5

Enter number of edges for node 4: 1

Enter edge 1 for node 4: 2

Enter number of edges for node 5: 0

Enter starting node for DFS: 1

DFS Traversal:

1 2 5 3 4

