DFS

def dfs(graph, start, visited=None):

if visited is None:

visited = set()

if start not in visited:

print(start, end=" ")

visited.add(start)

for neighbor in graph[start]:

dfs(graph, neighbor, visited)

# Get user input

graph = {}

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

for \_ in range(n):

node = input("Enter node: ")

neighbors = input(f"Enter neighbors of {node} (space separated): ").split()

graph[node] = neighbors

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

print("DFS Traversal:")

dfs(graph, start\_node)

OUTPUT

# Example Input:

# Enter number of nodes: 4

# Enter node: A

# Enter neighbors of A (space separated): B C

# Enter node: B

# Enter neighbors of B (space separated): D

# Enter node: C

# Enter neighbors of C (space separated): D

# Enter node: D

# Enter neighbors of D (space separated):

# Enter starting node for DFS: A

# Output:

# DFS Traversal:

# A B D C