def dls(graph, node, target, limit, visited=None):

if visited is None:

visited = set()

print(f"Visiting: {node}, Depth Limit: {limit}")

if node == target:

print("Target found!")

return True

if limit <= 0:

return False

visited.add(node)

for neighbor in graph[node]:

if neighbor not in visited:

if dls(graph, neighbor, target, limit - 1, visited):

return True

return False

# 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 start node: ")

target\_node = input("Enter target node to search: ")

depth\_limit = int(input("Enter depth limit: "))

print("\nDepth-Limited Search Trace:")

found = dls(graph, start\_node, target\_node, depth\_limit)

if not found:

print("Target not found within depth limit.")

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 start node: A

# Enter target node to search: D

# Enter depth limit: 2

# Example Output:

# Depth-Limited Search Trace:

# Visiting: A, Depth Limit: 2

# Visiting: B, Depth Limit: 1

# Visiting: D, Depth Limit: 0

# Target found!