

ASSIGNMENT 1

CODE:

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

    while queue:
        node = queue.pop(0)

        if node not in visited:
            visited.add(node)
            print(node, end=' ')

            for neighbor in graph[node]:
                if neighbor not in visited:
                    queue.append(neighbor)

def dfs(graph, start):
    visited = set()
    stack = [start]

    while stack:
        node = stack.pop()

        if node not in visited:
            visited.add(node)
            print(node, end=' ')

            for neighbor in graph[node]:
                if neighbor not in visited:
                    stack.append(neighbor)

def create_graph():
    key = int(input("Enter the number of nodes: "))
    graph = {}
    for i in range(1, key + 1):
        value = input(f"Enter the neighbouring nodes for node {i}: ")
        neighbors = list(map(int, value.split()))
        graph[i] = neighbors
    return graph

graph = create_graph()
print("The created graph is:", graph)

start_node = int(input("Enter the starting node for traversal: "))

print("BFS:")
bfs(graph, start_node)

print("\nDFS:")
dfs(graph, start_node)
```

OUTPUT:

Enter the number of nodes: 5

Enter the neighbouring nodes for node 1: 2 3 4

Enter the neighbouring nodes for node 2: 1 3 4 5

Enter the neighbouring nodes for node 3: 1 2

Enter the neighbouring nodes for node 4: 1 2

Enter the neighbouring nodes for node 5: 2

The created graph is: {1: [2, 3, 4], 2: [1, 3, 4, 5], 3: [1, 2], 4: [1, 2], 5: [2]}

Enter the starting node for traversal: 2

BFS:

2 1 3 4 5

DFS:

2 5 4 1 3