EXP : 2

1. Programs on Problem Solving

b. Solve any problem using depth first search.

AIM:

To solve any problem using depth first search.

CODE:

```
def dfs(node, graph, visited):
    # Mark the current node as visited and print it
    visited.add(node)
    print(node, end=' ')

    # Visit all the neighbors
    for neighbor in graph[node]:
        if neighbor not in visited:
            dfs(neighbor, graph, visited)

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

# Set to keep track of visited nodes
visited = set()
```

```
# Start DFS from node 'A'
print("DFS traversal starting from node 'A':")
dfs('A', graph, visited)
```

OUTPUT:

```
DFS traversal starting from node 'A':
A B D E F C
```

RESULT:

Thus the program is compiled and run successfully.