**Name: Kamran**

**Roll No: 002**

**Lab 5**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Program 3: DFS (Depth-First Search) Traversal using Stack and Node Class**

**What is this Program?**

This program performs a Depth-First Search (DFS) traversal on a tree or graph using a stack and Node objects. It visits nodes along a path deeply before backtracking.

**How does it work?**

Step 1: Initialize nodes

- Each node has a `value` and a list of `children`.

Step 2: Push start node to stack

- The starting node is added to the stack to begin traversal.

Step 3: Process nodes in the stack

- Pop the top node from the stack.

- Print its value.

- If it is the goal, stop traversal.

- Otherwise, push its unvisited children to the stack in reverse order.

Step 4: Repeat

- Continue until the stack is empty or the goal is found.

Output Example

**Example Run:**

DFS Stack Traversal Output: ['A', 'B', 'D', 'E', 'C']

Goal: C found!

Why this approach?

1. Depth-First Traversal: Explores nodes along paths deeply before backtracking.

2. Stack-Based Implementation: Simple way to avoid recursion.

3. Node Reusability: Node class allows building any tree or graph structure.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_