#### **Name: Javeria Manzoor**

**Roll no: BSAI-3C-201**

**Submitted to: Sir Rasikh**

**Lab task:05**

**Subject:Artificial Intelligence**

### **DFS with Stack (Graph Traversal)**

#### **How it works:**

1. The algorithm starts from a given start node (like 'A').
2. It uses a stack (LIFO – Last In, First Out) to keep track of nodes that need to be visited.
3. It repeatedly pops a node from the stack (takes the top element).
4. If that node hasn’t been visited yet, it:
5. Prints the node (to show it’s visited)
6. Adds it to the visited set.
7. its unvisited neighbors onto the stack (in reverse order to maintain correct DFS order).

**Why it works:**

* DFS goes deep first, exploring one branch fully before backtracking.
* Using a stack simulates recursion manually.
* The visited set avoids visiting the same node again (prevents infinite loops).

**QUESTION:2**

### **Preorder, Inorder, and Postorder Traversals (Tree Traversal)**

Here we treat the structure as a binary tree, where each node has:

* A left child
* A right child
* Preorder (Root → Left → Right)

**How:**

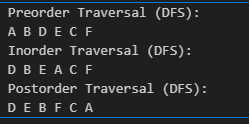
Visit the root first, then go left, then right.

Visit left child, then root, then right child.

Visit left subtree first, then right, then root last.  
**Why:**  
 Used when you want to process the parent before its children (e.g., copying a tree).

#### Inorder (Left → Root → Right)

#### .Postorder (Left → Right → Root)



### 

#### 