**Name: Kamran**

**Roll No: 002**

**Lab 6**

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

**Program 2: BFS (Breadth-First Search) Traversal using Node Class**

**What is this Program?**

This program performs a Breadth-First Search (BFS) traversal on a graph represented using `Node` objects. It visits all nodes level by level starting from a given node.

**How does it work?**

Step 1: Initialize nodes

- Each node has a `value`, a `visited` flag, and a list of `neighbors`.

Step 2: Mark start node visited and add to queue

- The starting node is marked visited and added to the BFS queue.

Step 3: Process nodes in the queue

- Remove the first node from the queue.

- Print its value.

- For each neighbor, if it hasn’t been visited, mark it visited and add it to the queue.

Step 4: Repeat

- Continue until the queue is empty.

Step 5: Reset visited nodes

- After BFS, all nodes’ `visited` flags are reset to allow future traversals.

Output Example

**Example Run:**

BFS traversal output: J E A C F G

**Why this approach?**

1. Level-wise Traversal: Visits nodes level by level.

2. Easy Implementation: Uses queue to manage the traversal.

3. Reusable: Node class allows building any graph structure.

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