

# Lab Assignment: Graph Traversal using Adjacency List

## Objective

To implement graph traversal algorithms using adjacency list representation (array of linked lists).

## Problem Statement

Write a C program to represent a graph using an **adjacency list** (array of linked lists) and perform both **Depth First Search (DFS)** and **Breadth First Search (BFS)** traversals. The graph is assumed to be undirected.

## Function Signatures

```
struct Node {
    int vertex;
    struct Node* next;
};

struct Graph {
    int numVertices;
    struct Node** adjLists; // Array of linked lists
    int* visited;
};

struct Graph* createGraph(int vertices);
struct Node* createNode(int v);
void addEdge(struct Graph* graph, int src, int dest);
void DFS(struct Graph* graph, int startVertex);
void BFS(struct Graph* graph, int startVertex);
```

## Details

- Create a graph using adjacency list representation. Each vertex has a linked list of all its adjacent vertices.
- Implement **DFS** using recursion.
- Implement **BFS** using a queue.
- Display the order of traversal for both DFS and BFS starting from a given vertex.

## Example Input

```
Enter number of vertices: 5
Enter number of edges: 4
Enter edges:
1 2
1 3
2 4
3 5
Enter start vertex: 1
```

## Expected Output

Adjacency List Representation:

1 -> 2 -> 3

2 -> 1 -> 4

3 -> 1 -> 5

4 -> 2

5 -> 3

DFS Traversal starting from vertex 1: 1 2 4 3 5

BFS Traversal starting from vertex 1: 1 2 3 4 5

## Learning Outcomes

- Represent a graph efficiently using adjacency lists.
- Implement and compare DFS (recursive) and BFS (iterative) traversals.
- Understand the role of stack/recursion and queue in graph exploration.