**Lab 15: Graphs**

**Task: Implement a Graph Data Structure using a 2D Array in C++**

**Description:**

In this lab task, you will implement a graph data structure using a 2D array. The program should allow the user to create a graph, add edges and vertices, and perform various operations on the graph, such as traversal and finding the shortest path.

Instructions:

1. Implement a class for the graph data structure using a 2D array representation.
2. The class should have functions for adding vertices, adding edges, and performing graph traversals.

.

Here are the necessary functions that the class should implement:

1. **Graph(int n)**: A constructor that creates an empty graph with n vertices.
2. **void addEdge(int from, int to, int weight)**: Adds an edge from vertex "from" to vertex "to" with a weight.
3. **void addVertex()**: Adds a new vertex to the graph.
4. **void bfsTraversal(int start):** Performs a breadth-first search traversal starting from the given vertex.
5. **void dfsTraversal(int start):** Performs a depth-first search traversal starting from the given vertex.
6. **void printGraph():** Prints the graph.

**Task: Implement a Graph Data Structure using a Node-based Approach (Adjacency list ) in C++**

In this lab, you will be implementing a graph data structure using a node-based implementation with an adjacency list. You will be using C++ for this lab.

To get started, create a **GraphNode** class that represents a node in the graph. This class should contain the following:

1. A data member that stores the ID of the node.
2. A data member that stores a list of the node's neighbors (i.e., nodes that are connected to this node).
3. A constructor that takes the ID of the node as an argument.
4. A method that adds a neighbor to the node.

Next, create a **Graph** class that represents the entire graph. This class should contain the following:

1. A data member that stores a list of all the nodes in the graph.
2. A method that adds a node to the graph.
3. A method that adds an edge between two nodes in the graph.
4. A method that prints out the adjacency list representation of the graph.
5. Your task is to implement the GraphNode and Graph classes as described above.