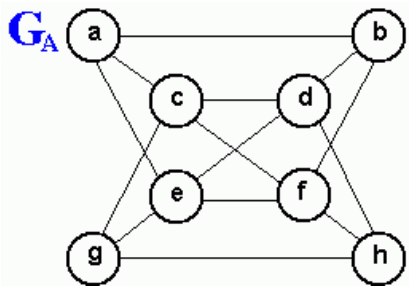
CL-## Data Structures – Lab

Problem#01: Adjacency Matrix –DFS 15 Marks

Consider the following undirected graph, store it in **adjacency matrix** first and do the followings.



A. **Check whether the given undirected graph contains a** return **True** if the graph contains any cycle and False otherwise.
B. Print the **In** and **Out** degree of each vertex?

Apply **Depth-first** traversal, starting from vertex **a**. Print the **order** in which the nodes are **visited** in a preorder traversal

- Apply depth-first traversal starting from vertex **a**.
- Apply depth-first traversal starting from vertex **d**.

Problem#02: Adjacency List – BFS 10 Marks

Implement an **adjacency list** to store the **graph of 7 vertices** given on right-side. perform an iterative-based **Breadth-first traversal** of this graph starting at vertex 0. Display the order of traversal on console.

```
0: 1 4
1: 5 0 3 6
2: 6 5
3: 4 1 5
4: 3 6 0
5: 1 2 3
6: 4 2 1
```

Problem#03: Adjacency Matrix - Dijkstra Algorithm 5 Marks

You are given a **weighted** graph representing a **road network**, where the weight of each edge represents the distance between two cities. Use adjacency matrix to store the below graph.

- Find the shortest path b/w any two cities and print those vertices.