Course Code	**:**	**MCSL-209**
Course Title	**:**	**Data Structures and Algorithms Lab**
Assignment Number	**:**	**BCA_NEW/BCAOL(III)L-209/Assign/2026-27**
Maximum Marks	**:**	**100**
Last Dates for Submission	**:**	**31st October, 2026 (For July 2026 Session)**
	:	**30th April, 2027 (For January 2027 Session)**

There are two questions in this assignment carrying a total of 40 marks. Each question carries 20 marks. Your Lab Record will carry 40 Marks. Rest 20 marks are for viva voce. You may use illustrations and diagrams to enhance the explanations. Please go through the guidelines regarding assignments given in the Programme Guide for the format of presentation.

Q1: Write an algorithm and program in 'C' language that accepts a Graph as input and prints its adjacency matrix and adjacency list.

Q2: Write an algorithm to list the data fields of the nodes of a Binary tree T by level. Within levels, nodes are to be listed left to right.

Q1: Write an algorithm and program in ‘C’ language that accepts a Graph as input and prints its adjacency matrix and adjacency list.

Answer

A graph G = (V, E) consists of vertices and edges joining pairs of vertices. It can be stored using an adjacency matrix or an adjacency list. The program below reads the number of vertices and the edge pairs, constructs both representations, and prints them. Vertices are numbered V0, V1, ... , V(n-1). It accepts either an undirected graph or a directed graph, as specified by the user.

Algorithm

Step A. Read the number n of vertices. Check that n is within the supported range. Read the number m of edges and whether the graph is directed or undirected.

Step B. Initialize an n by n adjacency matrix A with zeros.

Step C. For each input edge (u, v), check that both vertex numbers are valid and set A[u][v] = 1. For an undirected graph, also set A[v][u] = 1.

Step D. For each vertex i, scan row i of the completed matrix from column 0 to n - 1. For every A[i][j] = 1, append vertex j to the adjacency linked list of i.

Step E. Print the matrix row by row, followed by every linked list of neighbours. Finally, free the memory occupied by the linked lists.

Illustrative graph used for verification

For an undirected example, take V = {V0, V1, V2, V3, V4} and E = {(0,1), (0,2), (1,2), (1,3), (2,4), (3,4)}. The diagram is constructed directly from these six edge pairs. Each edge appears symmetrically in the matrix and in the corresponding neighbour lists.

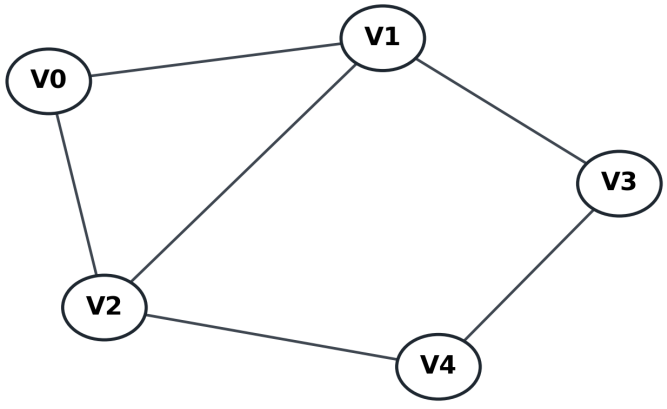


Figure 1. Five-vertex undirected graph used in the sample run.