University of Colombo School of Computing
SCS 1208 - Data Structures and Algorithms II

Lab Sheet 5

1) Write a C program to represent a graph using an adjacency list. The program should dynamically allocate memory for the graph and its components. The user should be able to input the number of vertices and edges, and then enter the details of each edge. After constructing the graph, print the adjacency list representation.

   Requirements:

   - Define a structure to represent a node in the adjacency list.
   - Define a structure to represent the graph, including the number of vertices, an array of pointers to adjacency list nodes, and any additional information you deem necessary.
   - Implement a function to create a graph by dynamically allocating memory for the required structures and adjacency list.
   - Implement a function to add an edge to the graph.
   - Implement a function to print the adjacency list representation of the graph.

2) Write a C program to create an undirected graph using an adjacency matrix. The program should do the following:

   - Prompt the user to enter the number of vertices and edges in the graph.
   - Dynamically allocate memory for an adjacency matrix based on the entered number of vertices.
   - Ask the user to input pairs of vertices representing edges in the graph.
   - Populate the adjacency matrix based on the entered edges.
   - Print the resulting adjacency matrix, representing the connections between vertices.
   - Display the total number of vertices and edges in the graph.

   Ensure that the program is straightforward and concise. Use dynamic memory allocation (e.g., malloc) to handle graphs of different sizes efficiently.

3) Code a function in C that frees the memory allocated for a graph and its components using malloc. Assume the graph has 3 vertices, and each vertex is connected to every other vertex. Test your function in the main program to avoid memory leaks.

4) Imagine you have a social network with three users: Alice, Bob, and Carol. Each user is connected to every other user in the network. Create a program to represent this as an undirected graph where each user is a vertex, and there is an edge between every pair of users.

Submit one file for all the questions by renaming your document as,
Index_Number_Practical_No.