- Assignment Name: Lab 8
- Description: Question based on Graphs
- Total Marks: 10
- Deadline: Depends on respective lab hours

- Problem Title: Shortest path in a graph from source to destination
- Marks Allotted: 10
- Description:

Given a directed graph, a source vertex, and a destination vertex, find the minimum distance taken from the source vertex to the destination vertex. If a path does not exist from the given source to the destination, the distance between them is -1.
Complete the find_dist function and print the answer.

A Graph is a non-linear data structure consisting of vertices and edges. More formally a Graph is composed of a set of vertices(V) and a set of edges(E). The edges form the connection between the vertices. The graph is denoted by G(E, V).

A graph can be represented as an adjacency matrix or an adjacency list. For this problem we will be going ahead with an adjacency matrix.

The input of the graph will be given in the form of edges between two vertices. The helper function create_graph has been provided, which will create an adjacency matrix representation for you.

You are required to complete the find_dist function, which finds the distance between source and destination vertices in the graph, using BFS(Breadth First Search) traversal technique.

- Int find_dist(Graph * adj_mat, int source, int dest)
- Takes in the adjacency matrix, source, and destination vertex, and prints the distance from the source to the destination vertex.

You are required to use the code provided to you as the boilerplate. The skeleton for the graph is provided to you along with the function to create the adjacency matrix.
Note that the given graph is directed.

```
7   typedef struct graph {
8   int n; /* Number of vertices in graph */
9   int adj[MAX_NODES][MAX_NODES]; /* Adjacency matrix */
10  } Graph;
```

Some helper functions to implement insertion and deletion in queue is also provided.

Using the boilerplate code is mandatory.

Input Format:

- The first line consists of an integer, denoting the number of vertices in the graph.
- The next few lines take in two integers, vertex1 and vertex2 separated by space, which has an edge between them. You can enter an unlimited number of edges until you enter a, -1 -1.
- The next two lines take in an integer value of the source vertex and destination vertex.

Output format:

- Print the distance from the source to the destination vertex.

Sample Test Cases:

1. TC#1
   - Input:
     8
     0 1
     0 3
     1 2
     3 4
     3 7
     4 5
     4 6
     4 7
     5 6
     6 7
     -1 -1
     0
     7
   - Output:
     2
   - Explanation:
     Shortest path from 0 to 7 is: 0 -> 3 -> 7
     Takes a distance of 2
2. TC#2
   - Input:
     6
     0 1
     0 5
     5 4
     4 3
     3 2

-1 -1
1
2

- Output:
  -1

- Explanation:
  There is no path from 1 to 2