PROGRAM 9:

a) Write a program to traverse a graph using BFS method.

b) Write a program to check whether given graph is connected or not using DFS method

```c
#include <stdio.h>
#include <stdlib.h>
#define MAX 100

void bfs(int graph[MAX][MAX], int visited[], int n, int start) {
    int queue[MAX], front = -1, rear = -1;
    printf("BFS Traversal starting from vertex %d: ", start);

    queue[++rear] = start;
    visited[start] = 1;

    while (front != rear) {
        int current = queue[++front];
        printf("%d ", current);

        for (int i = 0; i < n; i++) {
            if (graph[current][i] == 1 && !visited[i]) {
                queue[++rear] = i;
                visited[i] = 1;
            }
        }
    }
    printf("\n");
}

void dfs(int graph[MAX][MAX], int visited[], int n, int start) {
    visited[start] = 1;
    printf("%d ", start);
    for (int i = 0; i < n; i++) {
        if (graph[start][i] == 1 && !visited[i]) {
            dfs(graph, visited, n, i);
        }
    }
}
```

```c
int isConnected(int graph[MAX][MAX], int n) {
    int visited[MAX] = {0};

    dfs(graph, visited, n, 0);

    for (int i = 0; i < n; i++) {
        if (!visited[i]) {
            return 0;
        }
    }
    return 1;
}

int main() {
    int graph[MAX][MAX], visited[MAX], n, start;

    printf("Enter the number of vertices: ");
    scanf("%d", &n);

    printf("Enter the adjacency matrix of the graph:\n");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &graph[i][j]);
        }
    }

    printf("Enter the starting vertex for BFS: ");
    scanf("%d", &start);
    for (int i = 0; i < n; i++) {
        visited[i] = 0;
    }
    bfs(graph, visited, n, start);
    printf("--------END-------\n");

    printf("Enter the starting vertex for DFS: ");
    scanf("%d", &start);
    for (int i = 0; i < n; i++) {
        visited[i] = 0;
    }
```

```
    printf("DFS Traversal starting from vertex %d: ", start);
    dfs(graph, visited, n, start);
    printf("\n");

    if (isConnected(graph, n)) {
        printf("The graph is connected.\n");
    } else {
        printf("The graph is not connected.\n");
    }

    return 0;
}
```

OUTPUT

Output Clear

```
Enter the number of vertices: 4
Enter the adjacency matrix of the graph:
1 0 1 0
1 0 1 1
1 1 1 0
0 1 0 1
Enter the starting vertex for BFS: 3
BFS Traversal starting from vertex 3: 3 1 0 2
--------END-------
Enter the starting vertex for DFS: 2
DFS Traversal starting from vertex 2: 2 0 1 3
    --------END-------

0 2 1 3 The graph is connected.
```