Program 11 : Graph Reachability using DFS/BFS

Develop a Program in C for the following operations on Graph(G) of Cities

a. Create a Graph of N cities using Adjacency Matrix.

b. Print all the nodes reachable from a given starting node in a digraph using DFS/BFS method

```
#include <stdio.h>
#include <stdlib.h>

#define MAX 100  // Maximum number of cities

// Global Variables
int adjMatrix[MAX][MAX];  // Adjacency matrix to store the graph
int visited[MAX];         // Array to keep track of visited cities
int n;                    // Number of cities

// Function to create the graph
void createGraph() {
    int i, j, edges, origin, destin;

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

    // Initialize adjacency matrix to 0
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            adjMatrix[i][j] = 0;
        }
    }

    printf("Enter number of roads between the cities: ");
    scanf("%d", &edges);

    for (i = 0; i < edges; i++) {
        printf("Enter origin and destination cities numbers between 0 and %d: ",
n-1);
        scanf("%d %d", &origin, &destin);
        adjMatrix[origin][destin] = 1;
    }
}
```

```
// DFS function
void DFS(int city) {
    int i;
    printf("%d ", city);
    visited[city] = 1;

    for (i = 0; i < n; i++) {
        if (adjMatrix[city][i] == 1 && !visited[i]) {
            DFS(i);
        }
    }
}

// BFS function
void BFS(int startCity) {
    int queue[MAX], front = 0, rear = -1, i;
    visited[startCity] = 1;
    queue[++rear] = startCity;

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

        for (i = 0; i < n; i++) {
            if (adjMatrix[currentCity][i] == 1 && !visited[i]) {
                queue[++rear] = i;
                visited[i] = 1;
            }
        }
    }
}

// Main function
int main() {
    int choice, startCity;

    createGraph();

    printf("Enter the starting city number between 0 and %d: ", n-1);
    scanf("%d", &startCity);

    printf("\nChoose method to find reachable cities:\n");
    printf("1. Depth First Search (DFS)\n");
    printf("2. Breadth First Search (BFS)\n");
    scanf("%d", &choice);

    // Reset visited array
    for (int i = 0; i < n; i++) visited[i] = 0;

    if (choice == 1) {
        printf("Cities reachable from city %d using DFS:\n", startCity);
        DFS(startCity);
    } else if (choice == 2) {
        printf("Cities reachable from city %d using BFS:\n", startCity);
        BFS(startCity);
    } else {
        printf("Invalid choice!\n");
    }

    return 0;
}
```