

```

#include <iostream>
using namespace std;

int i, j, k, n, qu[10], front, visited[10];
int stk[10], top, visit1[10], visited1[10];

int main() {
    int m, edge;
    cout << "Enter number of vertices: ";
    cin >> n;
    cout << "Enter number of edges: ";
    cin >> m;
    cout << "\nEDGES:\n";

    // Define the cost array and initialize it to 0
    int cost[10][10] = {0};

    for (k = 1; k <= m; k++) {
        cin >> edge;
        i = edge / 10;
        j = edge % 10;
        // Assuming cost[i][j] and cost[j][i] represent the edges
        cost[i][j] = 1;
        cost[j][i] = 1;
    }

    // Display adjacency matrix
    cout << "The adjacency matrix of the graph is:\n";
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            cout << cost[i][j] << " ";
        }
        cout << endl;
    }

    int v;
    cout << "Enter initial vertex for BFS: ";
    cin >> v;
    cout << "The BFS of the Graph is:\n";
    cout << v << endl;
    visited[v] = -1;
    k = 0;
    front = 0; // Initialize front for BFS
    int rear = 0; // Initialize rear for BFS
    while (k < n) {
        for (j = 0; j < n; j++) {
            if (cost[v][j] != 0 && visited[j] != 1 && visit1[j] != 1) {
                visit1[j] = 1;
                qu[rear++] = j;
            }
        }
        v = qu[front++];
        cout << v << " ";
        k++;
        visit1[v] = 0;
    }
}

```

```

        visited[v] = 1;
    }

    cout << "\nEnter initial vertex for DFS: ";
    cin >> v;
    cout << "The DFS of the Graph is:\n";
    cout << v << " ";
    visited1[v] = 1;
    top = 0;
    stk[top++] = v;
    while (top > 0) {
        int flag = 0;
        v = stk[top - 1];
        for (j = 0; j < n; j++) {
            if (cost[v][j] != 0 && visited1[j] != 1 && visit1[j] != 1) {
                cout << j << " ";
                visited1[j] = 1;
                stk[top++] = j;
                flag = 1;
                break;
            }
        }
        if (flag == 0) {
            top--;
        }
    }

    return 0;
}

```