```
#include <iostream>
#include <cstdlib> // For exit()
using namespace std;

int cost[10][10], i, j, k, n, qu[10], front = 0, rear = 0, v, visit[10], visited[10];
int stk[10], top = 0, visit1[10], visited1[10];

int main() {
   cout << "### DSAL Practical No. 06 (C-13) ###" << endl;

   int m, ch;

   while (true) {
      cout << "\nMENU:\n";
      cout << "1. Create\n";
      cout << "2. Display Adjacency Matrix\n";
      cout << "3. BFS Traversal\n";
      cout << "4. DFS Traversal\n";
      cout << "5. Exit\n";
      cout << "Enter Your Choice: ";
      cin >> ch;

      if (ch == 1) {
         cout << "Enter number of vertices: ";
         cin >> n;
         cout << "Enter number of edges: ";
         cin >> m;

         // Initialize matrices
         for (i = 0; i < n; i++) {
            for (j = 0; j < n; j++) {
```

```
            cost[i][j] = 0;
        }
    }

    cout << "\nEnter the connecting edges (format: source destination):\n";
    for (k = 1; k <= m; k++) {
        cin >> i >> j;
        cost[i][j] = 1;
        cost[j][i] = 1; // Since it's an undirected graph
    }
}

else if (ch == 2) {
    cout << "\nThe 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;
    }
}

else if (ch == 3) { // BFS
    for (i = 0; i < n; i++) {
        visit[i] = 0;
        visited[i] = 0;
    }
    front = rear = 0;

    cout << "Enter initial vertex for BFS (0 to " << n - 1 << "): ";
    cin >> v;
```

```
    cout << "The BFS traversal of the graph is:\n";
    cout << v << " ";
    visited[v] = 1;
    k = 1;

    while (k < n) {
        for (j = 0; j < n; j++) {
            if (cost[v][j] != 0 && visited[j] != 1 && visit[j] != 1) {
                visit[j] = 1;
                qu[rear++] = j;
            }
        }
        if (front != rear) {
            v = qu[front++];
            cout << v << " ";
            k++;
            visit[v] = 0;
            visited[v] = 1;
        } else {
            break; // No more vertices to visit
        }
    }
    cout << endl;
}

else if (ch == 4) { // DFS
    for (i = 0; i < n; i++) {
        visit1[i] = 0;
        visited1[i] = 0;
    }
```

```
    top = 0;

    cout << "\nEnter initial vertex for DFS (0 to " << n - 1 << "): ";
    cin >> v;

    cout << "The DFS traversal of the graph is:\n";
    cout << v << " ";
    visited1[v] = 1;
    k = 1;

    while (k < n) {
        for (j = n - 1; j >= 0; j--) {
            if (cost[v][j] != 0 && visited1[j] != 1 && visit1[j] != 1) {
                visit1[j] = 1;
                stk[top++] = j;
            }
        }
        if (top != 0) {
            v = stk[--top];
            cout << v << " ";
            k++;
            visit1[v] = 0;
            visited1[v] = 1;
        } else {
            break; // No more vertices to visit
        }
    }
    cout << endl;
}

else if (ch == 5) {
```

```
            cout << "Exiting program.\n";
            exit(0);
        }

        else {
            cout << "Invalid choice. Please try again.\n";
        }
    }

    return 0;
}
```