**IAI : Exp-2(Inlab)**

**Name :** Keyur Patel **Roll No :** 16010421073  **Batch :** A-2

**Code:**

#include <stdio.h>

#include <stdlib.h>

//defining size of max nodes as 100

#define MAX\_NODES 100

int graph[MAX\_NODES][MAX\_NODES];

int visited[MAX\_NODES];

int num\_nodes, num\_edges;

void dfs(int node) {

    int i;

    printf("%d ", node);

    visited[node] = 1;

    for (i = 0; i < num\_nodes; i++)

{

        if (graph[node][i] == 1 && visited[i] == 0) {

            dfs(i);

        }

    }

}

int main() {

    int i, j, node1, node2;

    printf("Enter the number of nodes and edges: ");

    scanf("%d%d", &num\_nodes, &num\_edges);

    for (i = 0; i < num\_nodes; i++) {

        for (j = 0; j < num\_nodes; j++) {

            graph[i][j] = 0;

        }

        visited[i] = 0;

    }

    for (i = 0; i < num\_edges; i++) {

        printf("Enter two nodes that form an edge: ");

        scanf("%d%d", &node1, &node2);

        graph[node1][node2] = 1;

        graph[node2][node1] = 1;

    }

    int start\_node;

    printf("Enter the starting node: ");

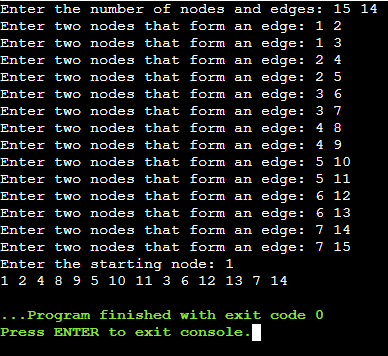
    scanf("%d", &start\_node);

    dfs(start\_node);

    return 0;

}

**Output:**

****

**Outcomes:**

**CO2 :** Analyze and formalize the problem (as a state space, graph, etc.) and select the appropriate search method and write the algorithm.

**Conclusions:**

Learnt to implement DFS - Uninformed search algorithm in state space with contents of Fringe and Visited nodes for each iteration of graphtraversal.