**Practical-13**

**AIM:** To implement Depth First Search in C language

**SOFTWARE REQUIRED:** Vs Code

**PSEUDO CODE:**

function DFS(graph, startVertex, visited):

if visited[startVertex] is false:

print startVertex

visited[startVertex] = true

for each neighbor in getNeighbors(startVertex):

if visited[neighbor] is false:

DFS(graph, neighbor, visited)

function getNeighbors(vertex):

// Helper function to return the neighbors of a vertex

neighbors = empty list

for each adjacentVertex in graph[vertex]:

if adjacentVertex is connected and not visited[adjacentVertex]:

add adjacentVertex to neighbors

return neighbors

**CODE:**

#include <stdio.h>

#include <stdlib.h>

#define MAX\_VERTICES 100

void DFS(int adjacencyMatrix[MAX\_VERTICES][MAX\_VERTICES], int vertices, int startVertex, int visited[MAX\_VERTICES]) {

    printf("%d ", startVertex);

    visited[startVertex] = 1;

    for (int i = 0; i < vertices; i++) {

        if (adjacencyMatrix[startVertex][i] && !visited[i]) {

            DFS(adjacencyMatrix, vertices, i, visited);

        }

    }

}

int main() {

    int vertices, startVertex;

    printf("Ananta Walli, A2305221322");

    printf("\nEnter the number of vertices: ");

    scanf("%d", &vertices);

    int adjacencyMatrix[MAX\_VERTICES][MAX\_VERTICES];

    int visited[MAX\_VERTICES] = {0};

    printf("Enter the adjacency matrix:\n");

    for (int i = 0; i < vertices; i++) {

        for (int j = 0; j < vertices; j++) {

            scanf("%d", &adjacencyMatrix[i][j]);

        }

    }

    printf("Enter the starting vertex: ");

    scanf("%d", &startVertex);

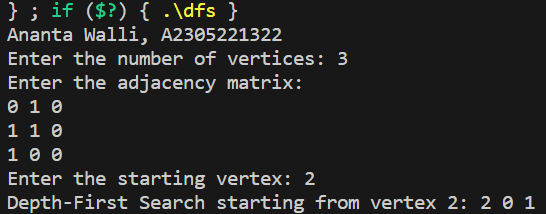
    printf("Depth-First Search starting from vertex %d: ", startVertex);

    DFS(adjacencyMatrix, vertices, startVertex, visited);

    return 0;

}

**OUTPUT:**



**RESULT:** The above code implements the Depth First Search in C programming.