EXERCISE-22

AIM: To write a C program to implement Depth-First Search (DFS) traversal of a graph using an adjacency matrix.

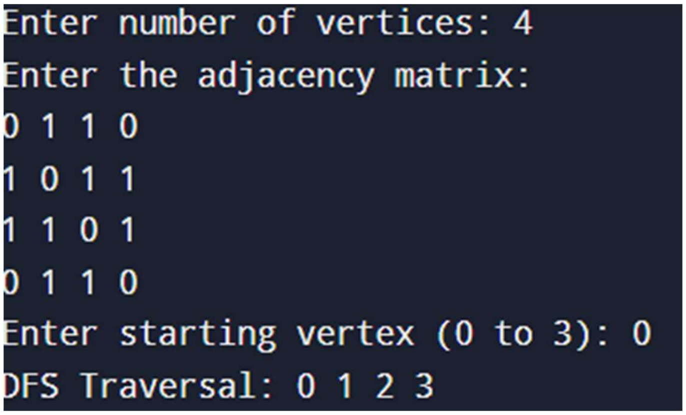
Algorithm:

1. Start.
2. Represent the graph using an adjacency matrix.
3. Use a visited[] array to track visited vertices.
4. Call the DFS(vertex) function:
5. Mark the current vertex as visited.
6. Print the vertex.
7. Recursively call DFS for all adjacent unvisited vertices.
8. End**.**

Program Code:

```
#include <stdio.h>
#define MAX 20
int visited[MAX];
void dfs(int adj[MAX][MAX], int n, int vertex) {
    printf("%d ", vertex);
    visited[vertex] = 1;
    for (int i = 0; i < n; i++) {
        if (adj[vertex][i] == 1 && !visited[i]) {
            dfs(adj, n, i);
        }
    }
}
int main() {
    int adj[MAX][MAX], n, start;
    printf("Enter number of vertices: ");
    scanf("%d", &n);
    printf("Enter the adjacency matrix:\n");
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            scanf("%d", &adj[i][j]);
    printf("Enter starting vertex (0 to %d): ", n - 1);
    scanf("%d", &start);
    for (int i = 0; i < n; i++)
        visited[i] = 0;
    printf("DFS Traversal: ");
    dfs(adj, n, start);
    printf("\n");
    return 0;
}
```

Input and Output:



```
Enter number of vertices: 4
Enter the adjacency matrix:
0 1 1 0
1 0 1 1
1 1 0 1
0 1 1 0
Enter starting vertex (0 to 3): 0
DFS Traversal: 0 1 2 3
```

Result:

The program successfully performs Depth-First Search (DFS) traversal of the given graph using recursion and an adjacency matrix.