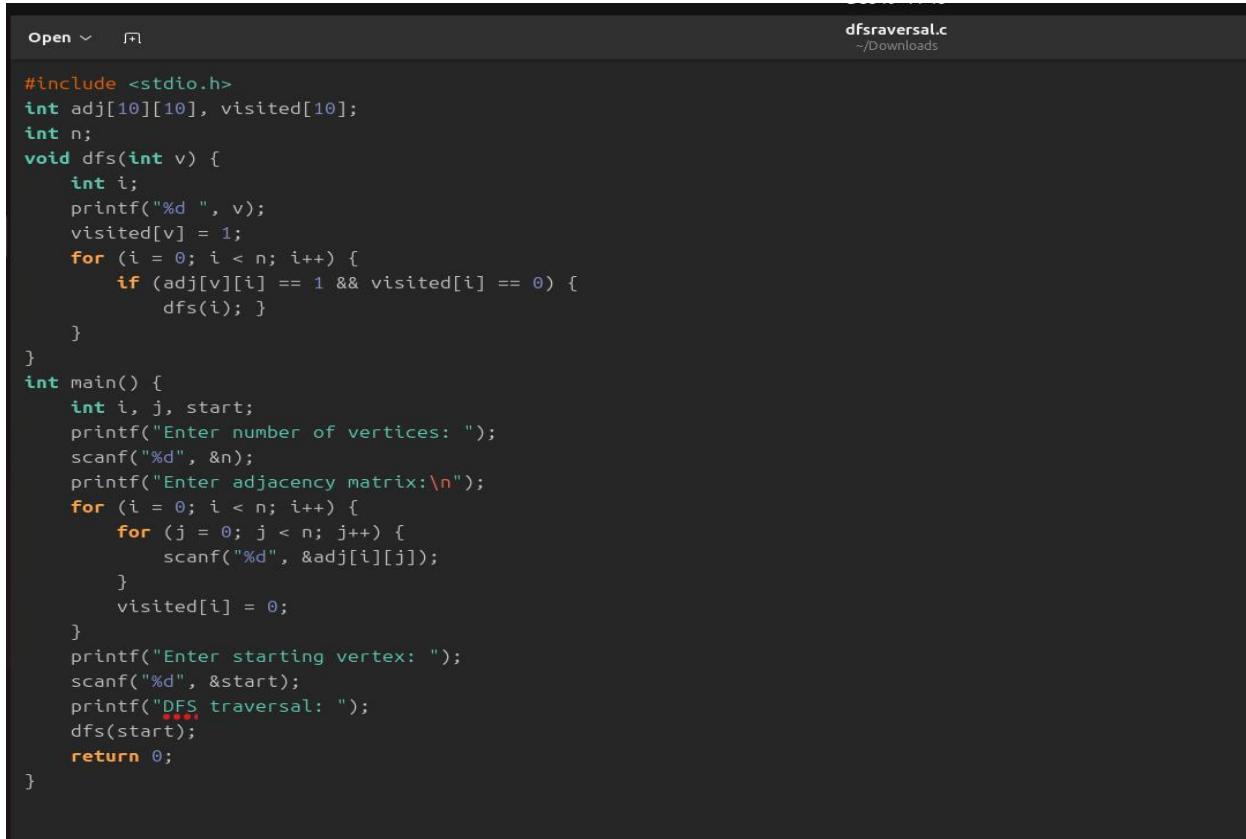


NAME : SAI LAKSHMI TEJASRI K

ROLL NO: CH.SC.U4CSE24243

1) DFS TRAVERSAL



The screenshot shows a code editor window with a dark theme. The file is named "dfsTraversal.c" located in the "/Downloads" directory. The code implements a Depth-First Search (DFS) algorithm. It starts by including the standard input-output library. It defines an adjacency matrix "adj" and an array "visited" to keep track of visited vertices. The "dfs" function takes a vertex index "v" as input and prints it. It then iterates through all vertices "i" from 0 to n-1. If vertex "i" is adjacent to "v" and has not been visited, it calls "dfs(i)". The "main" function prompts the user to enter the number of vertices and the adjacency matrix. It initializes the "visited" array and calls "dfs" starting from the vertex entered by the user.

```
#include <stdio.h>
int adj[10][10], visited[10];
int n;
void dfs(int v) {
    int i;
    printf("%d ", v);
    visited[v] = 1;
    for (i = 0; i < n; i++) {
        if (adj[v][i] == 1 && visited[i] == 0) {
            dfs(i);
        }
    }
}
int main() {
    int i, j, start;
    printf("Enter number of vertices: ");
    scanf("%d", &n);
    printf("Enter adjacency matrix:\n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            scanf("%d", &adj[i][j]);
        }
        visited[i] = 0;
    }
    printf("Enter starting vertex: ");
    scanf("%d", &start);
    printf("DFS traversal: ");
    dfs(start);
    return 0;
}
```



The screenshot shows a terminal window with a dark background. The user runs the compiled program "dfstraversal". They first enter the number of vertices as 4. Then they enter the adjacency matrix as a 4x4 grid of binary values. Finally, they enter the starting vertex as 0. The program outputs the DFS traversal path, which is 0 1 3 2.

```
amma@amma39:~/Downloads$ gcc dfstraversal.c -o dfstraversal
amma@amma39:~/Downloads$ ./dfstraversal
Enter number of vertices: 4
Enter adjacency matrix:
0 1 1 0
1 0 0 1
1 0 0 0
0 1 0 0
Enter starting vertex: 0
DFS traversal: 0 1 3 2 amma@amma39:~/Downloads$
```

2) BFS TRAVERSAL

```
Open ▾  bfstraversal.c
bfstraversal.c
~/Downloads

#include <stdio.h>
int adj[10][10], visited[10], queue[10];
int n, front = 0, rear = -1;
void bfs(int start) {
    int i, v;
    visited[start] = 1;
    queue[++rear] = start;
    while (front <= rear) {
        v = queue[front++];
        printf("%d ", v);
        for (i = 0; i < n; i++) {
            if (adj[v][i] == 1 && visited[i] == 0) {
                visited[i] = 1;
                queue[++rear] = i; }
        }
    }
}
int main() {
    int i, j, start;
    printf("Enter number of vertices: ");
    scanf("%d", &n);
    printf("Enter adjacency matrix:\n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            scanf("%d", &adj[i][j]); }
        visited[i] = 0; }
    printf("Enter starting vertex: ");
    scanf("%d", &start);
    printf("BFS traversal: ");
    bfs(start);
    return 0; }

amma@amma39:~/Downloads$ gcc bfstraversal.c -o bfstraversal
amma@amma39:~/Downloads$ ./bfstraversal
Enter number of vertices: 5
Enter adjacency matrix:
0 1 1 0 0
1 0 0 1 1
1 0 0 0 0
0 1 0 0 0
0 1 0 0 0
Enter starting vertex: 0
BFS traversal: 0 1 2 3 4 amma@amma39:~/Downloads$ █
```