1. C code implementation for BFS

Online Compiler

main.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#define MAX 100
int graph[MAX][MAX], visited[MAX], queue[MAX];
int front = 0, rear = 0;
void enqueue(int node) {
    queue[rear++] = node;}
int dequeue() {
    return queue[front++];}
bool isEmpty() {
    return front == rear;}
void bfs(int start, int n) {
    enqueue(start);
    visited[start] = 1;
    while (!isEmpty()) {
        int node = dequeue();
        printf("%d ", node);
        for (int i = 0; i < n; i++) {
            if (graph[node][i] && !visited[i]) {
                enqueue(i);
                visited[i] = 1;}}}}
int main() {
    int n, start;
    printf("Enter the 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", &graph[i][j]);
    printf("Enter the starting vertex: ");
    scanf("%d", &start);
    for (int i = 0; i < n; i++) visited[i] = 0;
    printf("BFS traversal: ");
    bfs(start, n);
    return 0;
}
```

Output

```
/tmp/IoaMsslnT9.o
Enter the number of vertices: 5
Enter the adjacency matrix:
0 1 1 0 0
1 0 1 1 0
1 1 0 1 1
0 1 1 0 1
0 0 1 1 00 1 1 0 0

1 0 1 1 0

1 1 0 1 1

0 1 1 0 1

0 0 1 1 0
Enter the starting vertex: 0
BFS traversal: 0 1 2 3 4

=== Code Execution Successful ===
```

2. C code implementation for DFS

main.c

```c
#include <stdio.h>
#define MAX 100
int graph[MAX][MAX];    // Adjacency matrix
int visited[MAX];
void dfs(int node, int n) {
    printf("%d ", node);
    visited[node] = 1;
    for (int i = 0; i < n; i++) {
        if (graph[node][i] && !visited[i]) {
            dfs(i, n);}}}
int main() {
    int n, start;
    printf("Enter the 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", &graph[i][j]);}}
    printf("Enter the starting vertex: ");
    scanf("%d", &start);
    if (start < 0 || start >= n) {
        printf("Invalid starting vertex\n");
        return -1;}
    for (int i = 0; i < n; i++) visited[i] = 0;
    printf("DFS traversal starting from vertex %d: ", start);
    dfs(start, n);
    printf("\n");
    return 0;
}
```

Output

```
/tmp/qevIA3W92n.o
Enter the number of vertices: 3
Enter the adjacency matrix:
0 1 3
0 2 5
0 6 9
Enter the starting vertex: 0
DFS traversal starting from vertex 0: 0 1 2


=== Code Execution Successful ===
```