

9B. b) Write a program to check whether given graph is connected or not using DFS method.

```
#include <stdio.h>
```

```
#define MAX 100
```

```
int visited[MAX];
```

```
// DFS function
```

```
void DFS(int graph[MAX][MAX], int n, int vertex) {  
    visited[vertex] = 1; // Mark current vertex as visited  
    for (int i = 0; i < n; i++) {  
        if (graph[vertex][i] == 1 && !visited[i]) {  
            DFS(graph, n, i);  
        }  
    }  
}
```

```
// Function to check if the graph is connected
```

```
int isConnected(int graph[MAX][MAX], int n) {  
    // Initialize visited array to 0  
    for (int i = 0; i < n; i++)  
        visited[i] = 0;  
  
    DFS(graph, n, 0); // Start DFS from vertex 0
```

```

// Check if all vertices are visited
for (int i = 0; i < n; i++) {
    if (!visited[i])
        return 0; // Graph is not connected
}
return 1; // Graph is connected
}

int main() {
    int n;
    int graph[MAX][MAX];

    printf("Enter number of vertices: ");
    scanf("%d", &n);

    printf("Enter adjacency matrix:\n");
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            scanf("%d", &graph[i][j]);

    if (isConnected(graph, n))
        printf("The graph is connected.\n");
    else
        printf("The graph is not connected.\n");

    return 0;
}

```

OUTPUT:

```
PS C:\Users\chait\OneDrive\Desktop\ds> cd "c:\Users
Enter number of vertices: 4
Enter adjacency matrix:
1 1 1 0
1 0 0 1
0 0 1 1
0 0 0 1
The graph is connected.
PS C:\Users\chait\OneDrive\Desktop\ds> cd "c:\Users
Enter number of vertices: 3
Enter adjacency matrix:
1 1 0
0 0 0
1 1 0
The graph is not connected.
PS C:\Users\chait\OneDrive\Desktop\ds> █
```