

LAB(16-12-2025)

DFS

INPUT

```
C:\Users\BMSECESE\Desktop>1BF24CS243>LAB(16-11-2025)DFS>C:\CODEc> -  
1 #include <stdio.h>  
2  
3 int graph[20][20], visited[20], n;  
4  
5 /* DFS function */  
6 void DFS(int v) {  
7     visited[v] = 1;  
8  
9     for (int i = 0; i < n; i++) {  
10         if ((graph[v][i] == 1) && !visited[i]) {  
11             DFS(i);  
12         }  
13     }  
14 }  
15  
16 int main() {  
17     int start = 0, connected = 1;  
18  
19     printf("Enter number of vertices: ");  
20     scanf("%d", &n);  
21  
22     printf("Enter adjacency matrix:\n");  
23     for (int i = 0; i < n; i++) {  
24         for (int j = 0; j < n; j++) {  
25             scanf("%d", &graph[i][j]);  
26         }  
27     }  
28  
29     /* Initialize visited array */  
30     for (int i = 0; i < n; i++) {  
31         visited[i] = 0;  
32     }  
33  
34     /* Start DFS from vertex 0 */  
35     DFS(start);  
36  
37     /* Check if all vertices are visited */  
38     for (int i = 0; i < n; i++) {  
39         if (!visited[i]) {  
40             connected = 0;  
41             break;  
42         }  
43     }  
44  
45     if (connected)  
46         printf("The graph is CONNECTED.\n");  
47     else  
48         printf("The graph is NOT CONNECTED.\n");  
49  
50 }  
51  
52 }
```

```
C:\Users\BMSECESE\Desktop>1BF24CS243>LAB(16-11-2025)DFS>C:\CODEc> -  
1 void DFS(int v) {  
2     visited[v] = 1;  
3  
4     for (int i = 0; i < n; i++) {  
5         if ((graph[v][i] == 1) && !visited[i]) {  
6             DFS(i);  
7         }  
8     }  
9 }  
10  
11 int main() {  
12     int start = 0, connected = 1;  
13  
14     printf("Enter number of vertices: ");  
15     scanf("%d", &n);  
16  
17     printf("Enter adjacency matrix:\n");  
18     for (int i = 0; i < n; i++) {  
19         for (int j = 0; j < n; j++) {  
20             scanf("%d", &graph[i][j]);  
21         }  
22     }  
23  
24     /* Initialize visited array */  
25     for (int i = 0; i < n; i++) {  
26         visited[i] = 0;  
27     }  
28  
29     /* Start DFS from vertex 0 */  
30     DFS(start);  
31  
32     /* Check if all vertices are visited */  
33     for (int i = 0; i < n; i++) {  
34         if (!visited[i]) {  
35             connected = 0;  
36             break;  
37         }  
38     }  
39  
40     if (connected)  
41         printf("The graph is CONNECTED.\n");  
42     else  
43         printf("The graph is NOT CONNECTED.\n");  
44  
45     return 0;  
46 }
```

OUTPUT

The screenshot shows a terminal window with the following text:

```
PS C:\Users\BPSCECSE> cd 'c:\Users\BPSCECSE\Desktop\1BF24CS243\LAB(16-11-2025) DFS\output'
PS C:\Users\BPSCECSE\Desktop\1BF24CS243\LAB(16-11-2025) DFS\output> & .\CODE.exe
● Enter number of vertices: 4
● Enter adjacency matrix:
0 1 1 0
1 0 1 1
1 1 0 1
0 1 1 0
The graph is CONNECTED.
PS C:\Users\BPSCECSE\Desktop\1BF24CS243\LAB(16-11-2025) DFS\output> cd 'c:\Users\BPSCECSE\Desktop\1BF24CS243\LAB(16-11-2025) DFS\output'
PS C:\Users\BPSCECSE\Desktop\1BF24CS243\LAB(16-11-2025) DFS\output> & .\CODE.exe
● Enter number of vertices: 4
● Enter adjacency matrix:
0 1 0 0
1 0 0 0
0 0 0 1
0 0 1 0
The graph is NOT CONNECTED.
PS C:\Users\BPSCECSE\Desktop\1BF24CS243\LAB(16-11-2025) DFS\output> [ ]
```