

**Write a program to check whether given graph is connected or not using DFS method.**

```
#include <stdio.h>

int visited[10], adj[10][10], n;

/* DFS function */
void dfs(int v)
{
    int i;
    visited[v] = 1;

    for (i = 0; i < n; i++)
    {
        if (adj[v][i] == 1 && !visited[i])
        {
            dfs(i);
        }
    }
}

int main()
{
    int i, j, connected = 1;

    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;  
}  
  
/* Start DFS from vertex 0 */  
dfs(0);  
  
/* Check if all vertices are visited */  
for (i = 0; i < n; i++)  
{  
    if (!visited[i])  
    {  
        connected = 0;  
        break;  
    }  
}  
  
if (connected)  
    printf("The graph is CONNECTED.\n");  
else  
    printf("The graph is NOT CONNECTED.\n");  
  
return 0;  
}
```

## OUTPUT:

The screenshot shows a terminal window within a code editor interface. The terminal tab is selected at the top. The command entered is `cd 'c:\Users\NISCHAL\OneDrive\Documents\Desktop\dsa' & .\DFS.exe`. The user is prompted to "Enter number of vertices: 4". They respond with "4". Then, they are prompted to "Enter adjacency matrix:". The user enters a 4x4 adjacency matrix:

```
0 1 1 0  
1 1 0 1  
1 0 1 1  
0 1 1 0
```

The terminal then outputs "The graph is CONNECTED." followed by a closing bracket.

```
PS C:\Users\NISCHAL\OneDrive\Documents\Desktop\dsa> cd 'c:\Users\NISCHAL\OneDrive\Documents\Desktop\dsa\output'&.\DFS.exe  
● Enter number of vertices: 4  
Enter adjacency matrix:  
0 1 1 0  
1 1 0 1  
1 0 1 1  
0 1 1 0  
The graph is CONNECTED.  
○ PS C:\Users\NISCHAL\OneDrive\Documents\Desktop\dsa\output>
```