

**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:

```
28     ... for (i = 0; i < n; i++)
29     ... {
30     ...     for (j = 0; j < n; j++)
31     ...     {
32     ...         scanf("%d", &adj[i][j]);
33     ...     }
34     ... }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\WISCHAL\OneDrive\Documents\Desktop\dsa> cd 'c:\Users\WISCHAL\OneDrive\Documents\Desktop\dsa\output'
PS C:\Users\WISCHAL\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\WISCHAL\OneDrive\Documents\Desktop\dsa\output> 
```