

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

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

```
#define MAX 10
```

```
int graph[MAX][MAX];
```

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

```
/* DFS function */
```

```
void dfs(int v, int n) {
```

```
    int i;
```

```
    visited[v] = 1;
```

```
    for (i = 0; i < n; i++) {
```

```
        if (graph[v][i] == 1 && !visited[i]) {
```

```
            dfs(i, n);
```

```
        }
```

```
    }
```

```
}
```

```
/* Main function */
```

```
int main() {
```

```
    int n, i, j;
```

```
    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", &graph[i][j]);
```

```
        }
```

```

        visited[i] = 0;
    }

    /* Start DFS from vertex 0 */
    dfs(0, n);

    /* Check if all vertices are visited */
    for (i = 0; i < n; i++) {
        if (!visited[i]) {
            printf("Graph is NOT connected\n");
            return 0;
        }
    }

    printf("Graph is CONNECTED\n");
    return 0;
}

```

OUTPUT:



```

Enter number of vertices: 3
Enter adjacency matrix:
2
3
1
4
5
6
7
8
9
Graph is NOT connected

Process returned 0 (0x0)   execution time : 10.359 s
Press any key to continue.

```