Depth First Search (DFS)

```
#include <stdio.h>

#define MAX 100

int visited[MAX];

// DFS function
void dfs(int adj[MAX][MAX], int n, int v) {
  visited[v] = 1;
  for (int i = 0; i < n; i++) {
    if (adj[v][i] == 1 && !visited[i]) {
      dfs(adj, n, i);
    }
  }
}

// Function to check connectivity
int isConnected(int adj[MAX][MAX], int n) {
  for (int i = 0; i < n; i++) visited[i] = 0;

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

  for (int i = 0; i < n; i++) {
    if (!visited[i]) return 0; // If any vertex not visited → Not connected
  }
  return 1;
}

int main() {
  int n, i, j;
  int adj[MAX][MAX];

  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]);
        }
    }

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

    return 0;
}
```

Output: