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

#define MAX_VERTICES 100

struct Graph {
   int V;
   int adjMatrix[MAX_VERTICES][MAX_VERTICES];
};

void initGraph(struct Graph *G, int V) {
   G->V = V;
   for (int i = 0; i < V; i++) {
      for (int j = 0; j < V; j++) {
         G->adjMatrix[i][j] = 0;
      }
   }
}

void addEdge(struct Graph *G, int src, int dest) {
   G->adjMatrix[src][dest] = 1;
   G->adjMatrix[dest][src] = 1; // If the graph is undirected
}

void DFS(struct Graph *G, int v, bool visited[]) {
   visited[v] = true;
   for (int i = 0; i < G->V; i++) {
      if (G->adjMatrix[v][i] && !visited[i]) {
         DFS(G, i, visited);
      }
   }
}

bool isConnected(struct Graph *G) {
   bool visited[MAX_VERTICES] = {false};
   DFS(G, 0, visited); // Start DFS from vertex 0
   for (int i = 0; i < G->V; i++) {
      if (!visited[i]) {
         return false; // If any vertex is not reachable, return false
      }
   }
   return true;
}

int main() {
   struct Graph G;
   int V = 5; // Number of vertices
   initGraph(&G, V);
```

```
    addEdge(&G, 0, 1);
    addEdge(&G, 0, 2);
    addEdge(&G, 1, 2);
    addEdge(&G, 3, 4);

    if (isConnected(&G)) {
        printf("The graph is connected.\n");
    } else {
        printf("The graph is not connected.\n");
    }

    return 0;
}
```

```
The graph is not connected.
```