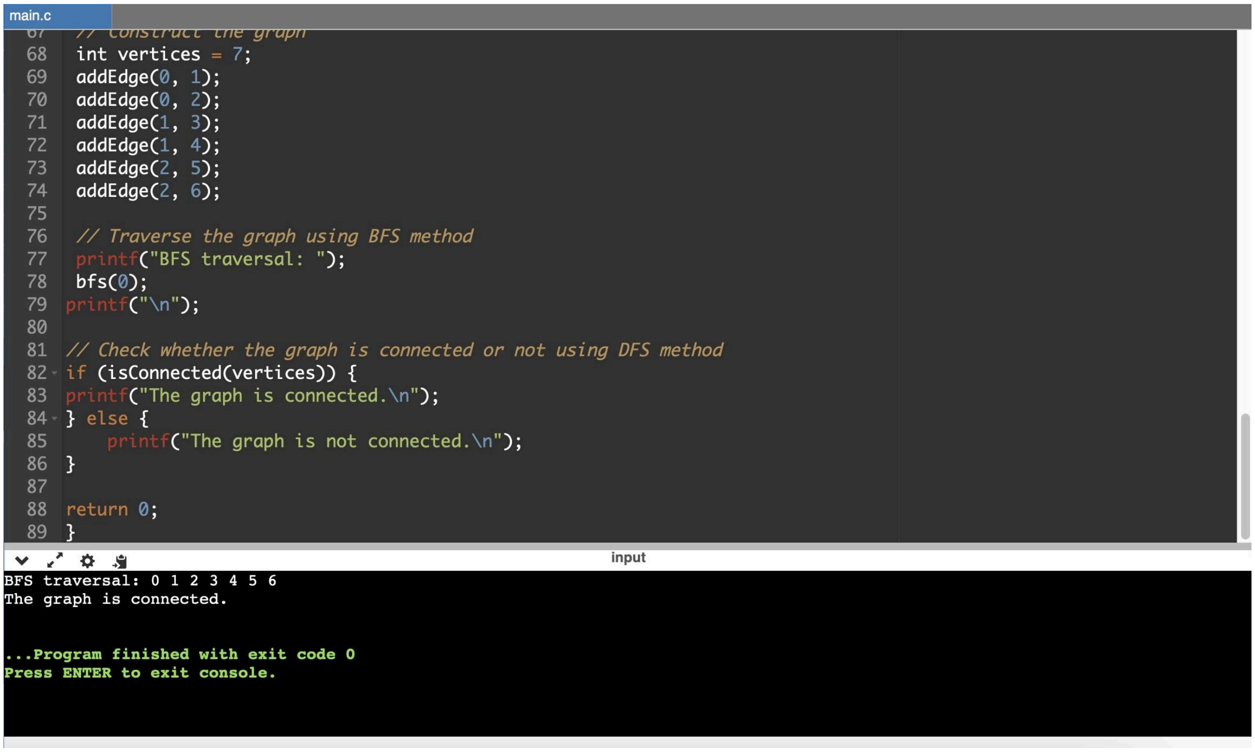
```
#include <stdio.h>
#include <stdlib.h>

// Define the maximum number of vertices in the graph
#define MAX_VERTICES 100

// Adjacency matrix representation of the graph
int graph[MAX_VERTICES][MAX_VERTICES];

// Function to add an edge between two vertices in the graph
void addEdge(int from, int to) {
    graph[from][to] = 1;
    graph[to][from] = 1;
}

// Function to perform BFS traversal of the graph
void bfs(int start) {
    int visited[MAX_VERTICES];
    for (int i = 0; i < MAX_VERTICES; i++) visited[i] = 0;
    int queue[MAX_VERTICES], front = 0, rear = 0;
    queue[rear++] = start;
    visited[start] = 1;
    while (front != rear) {
        int current = queue[front++];
        printf("%d ", current);
        for (int i = 0; i < MAX_VERTICES; i++) {
            if (graph[current][i] && !visited[i]) {
                queue[rear++] = i;
                visited[i] = 1;
            }
        }
    }
}

// Function to perform DFS traversal of the graph
void dfs(int start, int visited[], int* count) {
    visited[start] = 1;
    (*count)++;
    for (int i = 0; i < MAX_VERTICES; i++) {
        if (graph[start][i] && !visited[i]) {
            dfs(i, visited, count);
        }
    }
}
```

```
// Function to check whether the graph is connected or not
int isConnected(int vertices) {
    int visited[MAX_VERTICES];
    for (int i = 0; i < MAX_VERTICES; i++) visited[i] = 0;
    int count = 0;
    dfs(0, visited, &count);
    for (int i = 0; i < vertices; i++) {
        if (!visited[i]) return 0;
    }
    return 1;
}

int main() {
    // Construct the graph
    int vertices = 7;
    addEdge(0, 1);
    addEdge(0, 2);
    addEdge(1, 3);
    addEdge(1, 4);
    addEdge(2, 5);
    addEdge(2, 6);

    // Traverse the graph using BFS method
    printf("BFS traversal: ");
    bfs(0);
    printf("\n");

    // Check whether the graph is connected or not using DFS method
    if (isConnected(vertices)) {
        printf("The graph is connected.\n");
    } else {
        printf("The graph is not connected.\n");
    }

    return 0;
}
```

Output:



```c
 // Construct the graph
 int vertices = 7;
 addEdge(0, 1);
 addEdge(0, 2);
 addEdge(1, 3);
 addEdge(1, 4);
 addEdge(2, 5);
 addEdge(2, 6);

 // Traverse the graph using BFS method
 printf("BFS traversal: ");
 bfs(0);
printf("\n");

// Check whether the graph is connected or not using DFS method
if (isConnected(vertices)) {
printf("The graph is connected.\n");
} else {
    printf("The graph is not connected.\n");
}

return 0;
}
```

```
BFS traversal: 0 1 2 3 4 5 6
The graph is connected.


...Program finished with exit code 0
Press ENTER to exit console.
```