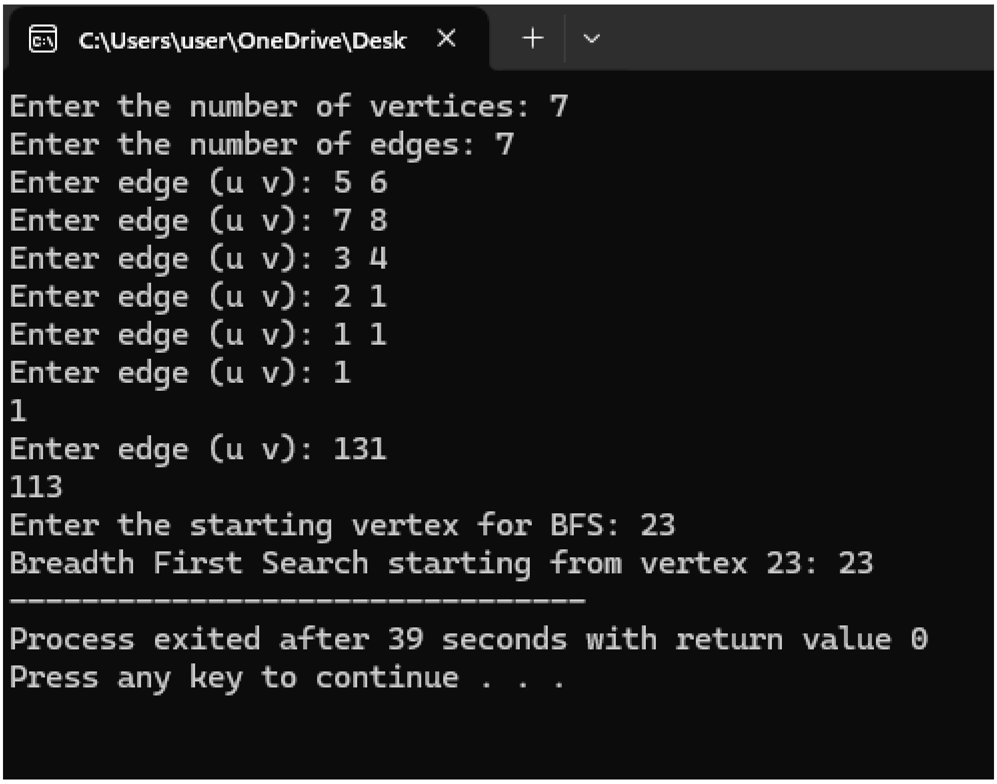
```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

#define MAX 100

int graph[MAX][MAX], visited[MAX], queue[MAX], front = -1, rear = -1;
int n;

void enqueue(int vertex) {
    if (rear == MAX - 1) {
        printf("\nQueue is full");
    } else {
        if (front == -1) {
            front = 0;
        }
        rear++;
        queue[rear] = vertex;
    }
}

int dequeue() {
    int vertex;
    if (front == -1) {
        printf("\nQueue is empty");
        return -1;
    } else {
        vertex = queue[front];
        front++;
        if (front > rear) {
            front = rear = -1;
        }
        return vertex;
    }
}

bool isEmpty() {
    return front == -1;
}

void bfs(int startVertex) {
    visited[startVertex] = 1;
    enqueue(startVertex);

    while (!isEmpty()) {
        int currentVertex = dequeue();
        printf("%d ", currentVertex);
        for (int i = 0; i < n; i++) {
            if (graph[currentVertex][i] == 1 && !visited[i]) {
                visited[i] = 1;
                enqueue(i);
            }
        }
    }
}

int main() {
    int edges, startVertex;

    printf("Enter the number of vertices: ");
    scanf("%d", &n);

    printf("Enter the number of edges: ");
    scanf("%d", &edges);

    for (int i = 0; i < edges; i++) {
        int u, v;
        printf("Enter edge (u v): ");
        scanf("%d %d", &u, &v);
        graph[u][v] = 1;
        graph[v][u] = 1; // For undirected graph
    }

    printf("Enter the starting vertex for BFS: ");
    scanf("%d", &startVertex);

    printf("Breadth First Search starting from vertex %d: ", startVertex);
    bfs(startVertex);

    return 0;
}
```



```
Enter the number of vertices: 7
Enter the number of edges: 7
Enter edge (u v): 5 6
Enter edge (u v): 7 8
Enter edge (u v): 3 4
Enter edge (u v): 2 1
Enter edge (u v): 1 1
Enter edge (u v): 1
1
Enter edge (u v): 131
113
Enter the starting vertex for BFS: 23
Breadth First Search starting from vertex 23: 23
--------------------------------
Process exited after 39 seconds with return value 0
Press any key to continue . . .
```