```
#include <stdio.h>
#include <stdbool.h>
#define MAX_VERTICES 100
int graph[MAX_VERTICES][MAX_VERTICES];
int visited[MAX_VERTICES];
int queue[MAX_VERTICES];
int front = 0, rear = 0;
void enqueue(int vertex) {
   queue[rear++] = vertex;
}
int dequeue() {
   return queue[front++];
}
bool isQueueEmpty() {
   return front >= rear;
}
void bfs(int startVertex, int numVertices) {
   visited[startVertex] = 1;
   enqueue(startVertex);
   printf("Breadth First Traversal starting from vertex %d: ", startVertex);
   while (!isQueueEmpty()) {
      int currentVertex = dequeue();
      printf("%d ", currentVertex);
      for (int i = 0; i < numVertices; ++i) {
         if (graph[currentVertex][i] && !visited[i]) {
            visited[i] = 1;
            enqueue(i);
         }
      }
   }
   printf("\n");
```

```
}
int main() {
    int numVertices, numEdges;
    printf("Enter the number of vertices: ");
    scanf("%d", &numVertices);
    for (int i = 0; i < numVertices; ++i) {
        visited[i] = 0;
        for (int j = 0; j < numVertices; ++j) {
            graph[i][j] = 0;
        }   }
    printf("Enter the number of edges: ");
    scanf("%d", &numEdges);
    for (int i = 0; i < numEdges; ++i) {
        int src, dest;
        printf("Enter edge %d (source destination): ", i + 1);
        scanf("%d %d", &src, &dest);
        graph[src][dest] = 1;
        graph[dest][src] = 1;    }
    int startVertex;
    printf("Enter the starting vertex for BFS: ");
    scanf("%d", &startVertex);
    bfs(startVertex, numVertices);
    return 0;
}
```

```
Enter the number of vertices: 6
Enter the number of edges: 5
Enter edge 1 (source destination): 1 2
Enter edge 2 (source destination): 1 3
Enter edge 3 (source destination): 3 4
Enter edge 4 (source destination): 3 5
Enter edge 5 (source destination): 4 6
Enter the starting vertex for BFS: 1
Breadth First Traversal starting from vertex 1: 1 2 3 4 5
```