```c
void BFS(struct Graph* graph, int startVertex) {
    int visited[graph->numVertices];
    for (int i = 0; i < graph->numVertices; i++) {
        visited[i] = 0;
    }
    int queue[graph->numVertices];
    int front = 0, rear = -1;
    visited[startVertex] = 1;
    queue[++rear] = startVertex;
    while (front <= rear) {
        int currentVertex = queue[front++];
        printf("%d ", currentVertex);
        struct Node* temp = graph->adjLists[currentVertex];
        while (temp != NULL) {
            int adjVertex = temp->dest;
            if (!visited[adjVertex]) {
                visited[adjVertex] = 1;
                queue[++rear] = adjVertex;
            }
            temp = temp->next;
        }
    }
}

int main() {
    int n, E, i, s, d,startVertex;
    printf("Enter no of vertices: ");
    scanf("%d", &n);
    printf("Enter no of edges: ");
    scanf("%d", &E);
    struct Graph* graph = createGraph(n);
    for (i = 1; i <= E; i++) {
        printf("Enter source: ");
        scanf("%d", &s);
        printf("Enter destination: ");
        scanf("%d", &d);
        addEdge(graph, s, d);
    }
    printf("Enter the starting vertex for BFS: ");
    scanf("%d", &startVertex);
    printf("Following is Breadth First Traversal (starting from vertex %d):\n", startVertex);
    BFS(graph, startVertex);

    return 0;
}
```

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

struct Node {
    int dest;
    struct Node* next;
};

struct Graph {
    int numVertices;
    struct Node** adjLists;
};

struct Node* newNode(int dest) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->dest = dest;
    newNode->next = NULL;
    return newNode;
}

struct Graph* createGraph(int numVertices) {
    struct Graph* graph = (struct Graph*)malloc(sizeof(struct Graph));
    graph->numVertices = numVertices;

    graph->adjLists = (struct Node**)malloc(numVertices * sizeof(struct Node*));

    for (int i = 0; i < numVertices; i++) {
        graph->adjLists[i] = NULL;
    }

    return graph;
}

void addEdge(struct Graph* graph, int src, int dest) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->dest = dest;
    newNode->next = graph->adjLists[src];
    graph->adjLists[src] = newNode;
}
```