```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;
}
```

```c
void DFSUtil(struct Graph* graph, int v, int visited[]) {
    visited[v] = 1;
    printf("%d ", v);

    struct Node* temp = graph->adjLists[v];
    while (temp != NULL) {
        int adj = temp->dest;
        if (!visited[adj]) {
            DFSUtil(graph, adj, visited);
        }
        temp = temp->next;
    }
}

void DFS(struct Graph* graph, int v) {
    int *visited = (int*)malloc(graph->numVertices * sizeof(int));
    for (int i = 0; i < graph->numVertices; i++) {
        visited[i] = 0;
    }

    DFSUtil(graph, v, visited);
}

int main() {
    int n, E, i, s, d;

    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("Following is Depth First Traversal (starting from vertex 0):\n");
    DFS(graph, 0);

    return 0;
}
```