#include<iostream>

#include<stdio.h>

#include<stdlib.h>

using namespace std;

struct node{

int vertex;

struct node \*next;

};

struct node \*createnode(int v);

struct Graph {

int numvertices;

int \*visited;

struct node\*\* addlist;

};

struct Graph\* creategraph(int);

void addedge(struct Graph\*,int,int);

void DFS(struct Graph\*,int);

int main(){

int no,src,ed,a,b;

cout << "\n Enter no of vertices edges and source node : ";

cin >> no >> ed >> src;

cout << no << " " << ed << " " << src;

struct Graph\* graph=creategraph(no);

for(int i=0;i<no;i++){

cout << "\n Enter source and destination : ";

cin >> a >> b;

addedge(graph,a,b);

}

DFS(graph,src);

return 0;

}

void DFS(struct Graph\* graph,int vertex){

struct node\* addlist=graph->addlist[vertex];

struct node\* temp=addlist;

graph->visited[vertex]=1;

cout << vertex << " ";

while(temp!=NULL){

int connectedvertex=temp->vertex;

if(graph->visited[connectedvertex]==0){

DFS(graph,connectedvertex);

}

temp=temp->next;

}

}

struct node\* createnode(int v){

struct node\* newnode=(struct node \*)malloc(sizeof(struct node));

newnode->vertex=v;

newnode->next=NULL;

return newnode;

};

struct Graph\* creategraph(int vertices){

struct Graph\* graph=(struct Graph \*)malloc(sizeof(struct Graph));

graph->numvertices=vertices;

graph->visited=(int \*)malloc(vertices\*sizeof(int));

graph->addlist=(struct node \*\*)malloc(vertices\*sizeof(struct node));

for(int i=0;i<vertices;i++){

graph->addlist[i]=NULL;

graph->visited=0;

}

return graph;

};

void addedge(struct Graph\* graph,int src,int dest){

struct node\* newnode=createnode(dest);

newnode->next=graph->addlist[src];

graph->addlist[src]=newnode;

// Add edge from dest to src

/\*

newNode = createNode(src);

newNode->next = graph->adjLists[dest]; // for making bidirectional graph

graph->adjLists[dest] = newNode;

\*/

}