***Depth First Search (DFS):***

#include<iostream>

#include<vector>

#include<string.h>

using namespace std;

vector<int>ver,adj[100];

bool flag[200];

void dfs(int source)

{

int stack[100];

int top=0;

stack[top] = source;

flag[source] = true;

while(top>=0)

{

int u = stack[top];

bool isTaken = false;

for(int i=0;i<adj[u].size();i++)

{

int v = adj[u][i];

if(!flag[v])

{

flag[v] = true;

top++;

stack[top] = v;

isTaken = true;

break;

}

}

if(!isTaken)

{

cout<<stack[top]<<" ";

top--;

}

}

cout<<endl;

}

int main()

{

int m, n, u, v, source;

while(cin>>m)

{

memset(flag,false,sizeof(flag));

ver.clear();

for(int i=0;i<m;i++)

{

cin>>u;

cin>>n;

ver.push\_back(u);

adj[u].clear();

for(int j=0;j<n;j++)

{

cin>>v;

adj[u].push\_back(v);

}

}

cin>>source;

dfs(source);

}

}