求满足（v能到达的点w都能到达v）的点的集合，即求强连通分量，最后输出出度为零的点

#include<iostream>

#include<stack>

using namespace std;

const int MAX=5002;

int graph[MAX][MAX];//邻接表存储图

int pre[MAX],low[MAX],id[MAX];

bool out[MAX];//记录点所在的连通分量号

int dindex=0;//记时间戳用

int sccn=0,n,m;//记连通分量个数

stack<int>s;

void Tarjan(int u)

{

int v,i;

int minx=pre[u]=low[u]=dindex++;//为节点i设定次序编号和low初值

s.push(u);//将节点压入栈中

for(i=1;i<=graph[u][0];i++)//枚举每一条边

{

v=graph[u][i];

if(pre[v]==-1)

Tarjan(v);

if(low[v]<minx)

minx=low[v];

}

if(minx<low[u])

{

low[u]=minx;

return;

}

sccn++;

do

{

low[v=s.top()]=m;//将v退栈，为强连通分量中的一个顶点

s.pop();

id[v]=sccn;

}while(v!=u);

}

void Component()

{

sccn=0,dindex=0;

memset(pre,-1,sizeof(pre));

for(int i=1;i<=m;i++)

if(pre[i]==-1)

Tarjan(i);

}

void dfs(int u)

{

pre[u]=1;

for(int i=1;i<=graph[u][0];i++)

{

int v=graph[u][i];

if(id[v]!=id[u])

out[id[u]]=true;

if(!pre[v])

dfs(v);

}

}

void DFS()

{

memset(out,false,sizeof(out));

memset(pre,0,sizeof(pre));

for(int i=1;i<=m;i++)

if(!pre[i])

dfs(i);

}

int main()

{

while(scanf("%d",&m)!=EOF&&m)//n表示边的个数，m表示点的个数

{

scanf("%d",&n);

int i,from,to;

for(i=1;i<=m;i++)//此处一定不能用memset赋值！！会超内存！

graph[i][0]=0;

for(i=0;i<n;i++)

{

scanf("%d%d",&from,&to);

graph[from][++graph[from][0]]=to;

}

Component();

DFS();

for(i=1;i<=m;i++)

if(!out[id[i]])

printf("%d ",i);

printf("\n");

}

return 0;

}