#include<stdio.h>

#include<stdlib.h>

int min(int a,int b)

{

return(a<b?a:b);

}

struct node

{

int val;

struct node\* next;

};

struct graph

{

int v;

struct node\*\* arr;

};

struct graph\* createGraph(int v)

{

int i;

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

temp->v=v;

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

temp->arr=(int\*\*)malloc(sizeof(int\*)\*v);

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

temp->arr[i]=NULL;

return temp;

}

void addEdge(struct graph\* g,int u,int v)

{

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

temp->val = v;

temp->next = g->arr[u];

g->arr[u] = temp;

}

void apUtil(struct graph \* g,int node,int\* isVisited,int\* des,int\* parent,int\* low,int\* ap)

{

struct node\* temp=NULL;

static int time=0;

int children=0;

temp = g->arr[node];

isVisited[node]=1;

time++;

//printf("\nsetting time for %d",node);

des[node]=low[node]=time;

while(temp!=NULL)

{

if(!isVisited[temp->val])

{

children++;

parent[temp->val]=node;

apUtil(g,temp->val,isVisited,des,parent,low,ap);

low[node]= min(low[node],low[temp->val]);

if(parent[node]==-1 && children>1)

ap[node]=1;

if(parent[node]!=-1 && des[node]<=low[temp->val])

ap[node]=1;

}

else if(temp->val!=parent[node])

{

low[node]=min(low[node],des[temp->val]);

}

temp= temp->next;

}

//printf("%d",node);

}

void AP(struct graph\* g)

{

int i;

int\* des = (int\*)malloc(sizeof(int)\*g->v);

int\* isVisited = (int\*)malloc(sizeof(int)\*g->v);

int\* parent = (int\*)malloc(sizeof(int)\*g->v);

int\* low = (int\*)malloc(sizeof(int)\*g->v);

int\* ap = (int\*)malloc(sizeof(int)\*g->v);

for(i=0;i<g->v;i++)

{

isVisited[i]=0;

parent[i]=-1;

ap[i]=0;

}

for(i=0;i<g->v;i++)

{

if(isVisited[i]==0)

{

apUtil(g,i,isVisited,des,parent,low,ap);

}

}

printf("\n");

int flag=0;

printf("Searching Articulation points ....");

for(i=0;i<g->v;i++)

{

if(ap[i]==1)

{

flag=1;

printf("\nVertex : %d",i);

}

}

if(flag==0)

printf("\nNo Articulation point found.");

}

void main()

{

int size=0,edges=0,i,u,v;

printf("Please enter size of the graph : ");

scanf("%d",&size);

printf("\nPlease number of edges : ");

scanf("%d",&edges);

struct graph\* g = createGraph(size);

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

{

printf("\nEnter the edge %d\n",i);

scanf("%d %d",&u,&v);

addEdge(g,u,v);

}

AP(g);

}