#include<iostream>

using namespace std;

int n,m;//n为点的个数，m为边的个数

const int MAX=10002;

struct Node

{

int to;

Node\* next;

~Node(){delete next;}

};

Node\* map[MAX];

Node\* reverse\_map[MAX];

int f[MAX],id[MAX];//out记录连通分量的出度，id记点属于哪个连通分量，f记录点深搜的完成时间

bool used[MAX],out[MAX];

int sccn,ftime;

void dfs1(int u)//第一次深搜形成f数组

{

used[u]=true;

for(Node \*t=map[u];t;t=t->next)

if(!used[t->to])

dfs1(t->to);

f[++ftime]=u;

}

void dfs2(int u)//按f递减的顺序对转置图进行深搜

{

used[u]=true;

id[u]=sccn;

for(Node \*t=reverse\_map[u];t;t=t->next)

{

if(!used[t->to])

dfs2(t->to);

if(id[t->to]&&id[t->to]!=sccn)

out[id[t->to]]++;

}

}

void Kosaraju()

{

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

memset(id,0,sizeof(id));

int i;

sccn=0,ftime=0;

memset(used,false,sizeof(used));

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

if(!used[i])

dfs1(i);

memset(used,false,sizeof(used));

for(i=ftime;i>0;i--)

if(!used[f[i]])

{

++sccn;

dfs2(f[i]);

}

}

int main()

{

while(scanf("%d%d",&n,&m)!=EOF)

{

memset(map,0,sizeof(map));

memset(reverse\_map,0,sizeof(reverse\_map));

int from,to,i;

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

{

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

Node\* temp=new Node;

temp->to=to;

temp->next=map[from];

map[from]=temp;

Node\* t=new Node;

t->to=from;

t->next=reverse\_map[to];

reverse\_map[to]=t;

}

Kosaraju();

int res=0,temp=-1;

bool flag=true;

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

{

if(!out[id[i]])

{

res++;

if(temp==-1)

temp=id[i];

else if(temp!=id[i])

flag=false;

}

}

if(flag)

printf("%d\n",res);

else

printf("%d\n",0);

}

}