#include<bits/stdc++.h>

using namespace std;

vector<vector<int> > graph;

void bridge(int node, int parent, vector<int> &visited, vector<int> &tin,

vector<int> &low, int &timer, vector<int> &articulation){

visited[node] = 1;

tin[node] = timer;

low[node] = timer;

timer++;

int child = 0;

for(auto x: graph[node]){

if(x==parent) continue;

if(!visited[x]){

bridge(x,node,visited,tin,low,timer,articulation);

low[node] = min(low[x],low[node]);

child++;

if(low[x]>=tin[node] && parent!=-1){

articulation[node] = 1;

}

}

else{

low[node] = min(low[node],tin[x]);

}

}

if(child>1 && parent==-1){

articulation[node] = 1;

}

}

int main(){

int n,m;

cin>>n>>m;

graph.resize(n+1);

while(m--){

int x,y;

cin>>x>>y;

graph[x].push\_back(y);

graph[y].push\_back(x);

}

vector<int> visited(n+1,0);

vector<int> tin(n+1,-1);

vector<int> low(n+1,-1);

vector<int> articulation(n+1,0);

int timer = 0;

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

if(!visited[i]){

bridge(i,-1,visited,tin,low,timer,articulation);

}

}

for(int i=1;i<articulation.size();i++){

if(articulation[i]==1)

cout<<i<<endl;

}

return 0;

}