#include<bits/stdc++.h>

using namespace std;

vector<int> par;

int find(int x){

if(par[x]==x){

return x;

}

return par[x] = find(par[x]);

}

void union\_(int x, int y){

x = find(x);

int yy = find(y);

if(yy!=y){

return;

}

par[yy] = x;

// if(x!=y){

// par[y] = x;

// }

}

int findMotherVertex(int V, vector<int>adj[])

{

par.resize(V);

for(int i=0;i<V;i++){

par[i] = i;

}

for(int i=0;i<V;i++){

int src = i;

for(int j=0;j<adj[src].size();j++){

int dest = adj[i][j];

union\_(src,dest);

// if(find(src)!=find(dest)){

// union\_(src,dest);

// }

}

}

int parentOfall = find(0);

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

if(find(i)!=parentOfall){

return -1;

}

}

return parentOfall;

}

int main(){

int n,e;

cin>>n>>e;

vector<int> adj[n];

for(int i=0;i<e;i++){

int x,y;

cin>>x>>y;

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

}

cout<<findMotherVertex(n,adj);

}