#include <iostream>

#include<bits/stdc++.h>

using namespace std;

// printing a undirected graph in bfs form

bool arr[100000];

long dist[100000];

vector<vector<int>>g;

void edge(int a,int b)

{

g[a].push\_back(b);

g[b].push\_back(a);

}

long bfs(int s,int v)

{

queue<int>q;

q.push(s);

dist[s]=0;

arr[s]=true;

while(!q.empty())

{

int x=q.front();

cout<< x<< " ";

q.pop();

// vector<int>::iterator it;

for(int i=0;i<g[x].size();i++)

{

if(arr[g[x][i]]==false)

{

dist[g[x][i]]=dist[x]+1;

q.push(g[x][i]);

arr[g[x][i]]=true;

}

}

}

return(dist[v]);

}

int main()

{

for(int i=0;i<100000;i++)

{

arr[i]=false;

}

for(int i=0;i<100000;i++)

{

dist[i]=-1;

}

int n, e;

cin >> n >> e;

//v.assign(n, false);

g.assign(n+1, vector<int>());

int a, b;

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

cin >> a >> b;

edge(a, b);

}

/\*for (int i = 0; i < n; i++) {

if (!arr[i])

bfs(i);

}\*/

long d=bfs(0,7);

cout<<endl<<d<<endl;

return 0;

}