#include<bits/stdc++.h>

using namespace std;

vector<vector<int> > graph;

void shortest\_path\_bfs(int src, vector<int> &distance){

queue<int> q;

q.push(src);

int d = 0;

distance[src] = d;

while(!q.empty()){

int node = q.front();

q.pop();

d = distance[node] + 1;

for(auto x: graph[node]){

if(distance[x]>d){

q.push(x);

distance[x] = d;

}

}

}

}

int main(){

int n;

int e;

cin>>n>>e;

graph.resize(n);

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

int x,y;

cin>>x>>y;

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

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

}

int src;

cin>>src;

vector<int> distance(n,INT\_MAX);

shortest\_path\_bfs(src,distance);

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

cout<<i<<" "<<distance[i]<<endl;

}

return 0;

}