#include<bits/stdc++.h>

using namespace std;

vector<int> parent(10000);

vector<int> rank\_(10000);

bool sameset(int a,int b){

int x = find\_par(a);

int y = find\_par(b);

return x==y;

}

int findPar(int node){

if(node==parent[node]){

return node;

}

parent[node] = findPar(parent[node]);

return parent[node];

}

void union\_(int u,int v){

int x = findPar(u);

int y = findPar(v);

if(rank\_[x]<rank\_[y]){

parent[x] = y;

}

else if(rank\_[y]<rank\_[x]){

parent[y] = x;

}

else{

parent[y] = x;

rank\_[x]++;

}

}

int main(){

int m;

int n;

cin>>n>>m;

// parent.resize(m+1);

// rank\_.resize(m+1);

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

parent[i] = i;

rank\_[i] = 0;

}

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

cout<<parent[i]<<" ";

}

cout<<endl;

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

cout<<rank\_[i]<<" ";

}

cout<<endl;

while(m--){

int x,y;

cin>>x>>y;

union\_(x,y);

}

int a,b;

cin>>a>>b;

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

cout<<parent[i]<<" ";

}

cout<<endl;

if(findPar(a)!=findPar(b)){

cout<<"not same";

}

else{

cout<<"Same";

}

}