#include<bits/stdc++.h>

using namespace std;

vector<vector<pair<int,int> > > graph;

void addEdge(int x, int y, int w){

graph[x].push\_back(make\_pair(y,w));

graph[y].push\_back(make\_pair(x,w));

}

void display(){

for(int i=0;i<graph.size();i++){

cout<<i<<"->";

for(auto it: graph[i]){

cout<<it.first<<" "<<it.second<<", ";

}

cout<<endl;

}

}

void hamilton(int ini, int src, vector<bool> visited, string ans, int count){

if(count==visited.size()){

for(auto it: graph[ini]){

if(src==it.first){

ans = ans+"\*";

cout<<ans<<endl;

return;

}

}

cout<<ans<<"."<<endl;

return;

}

visited[src] = true;

for(auto it: graph[src]){

if(visited[it.first]==false){

string str = to\_string(it.first);

hamilton(ini,it.first,visited,ans+str,count+1);

}

}

visited[src] = false;

}

int main(){

int v,e;

cin>>v>>e;

graph.resize(v);

while(e--){

int x,y,w;

cin>>x>>y>>w;

addEdge(x,y,w);

}

int src,dest;

cin>>src;

int ini = src;

// display();

vector<bool> visited(v);

string s = to\_string(src);

hamilton(ini,src,visited,s,1);

}