#include<bits/stdc++.h>

using namespace std;

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

void topo\_dfs(int src, stack<int> &st, vector<bool> &visited){

visited[src] = true;

for(auto x: graph[src]){

if(!visited[x.first])

topo\_dfs(x.first,st,visited);

}

st.push(src);

return;

}

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

stack<int> st;

vector<bool> visited(graph.size(),false);

topo\_dfs(src,st,visited);

// mnaking src distance[src] = 0 and poping it from the stack

distance[src] = 0;

while(!st.empty()){

int node = st.top();

st.pop();

if(distance[node]!=INT\_MAX){

for(auto x: graph[node]){

if(distance[x.first]>distance[node]+x.second){

distance[x.first] = distance[node] + x.second;

}

}

}

}

}

int main(){

int n;

int e;

cin>>n>>e;

graph.resize(n);

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

int x,y,w;

cin>>x>>y>>w;

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

}

int src;

cin>>src;

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

shortest\_path\_directed(src,distance);

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

if(distance[i]==INT\_MAX){

cout<<i<<" inf"<<endl;

}

else{

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

}

}

return 0;

}