//implementataion of weighted directed graph using adjacency list

#include<bits/stdc++.h>

using namespace std;

class edge

{

int wt;

int node;

public:

edge(int a,int b)

{

wt=a;node=b;

}

int retwt()

{

return(wt);

}

int retnode()

{

return(node);

}

};

int main()

{

int n,e;

cin>>n>>e;

vector<list<edge>>v(n);

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

{

int a,b;

cin>>a>>b;

int weight;

cin>>weight;

v[a].push\_back(edge(weight,b));

}

vector<list<edge>>::iterator it;

list<edge>::iterator it2;

for(it=v.begin();it!=v.end();it++)

{

// cout<<"neighbours of"<<\*it;

list<edge>temp=\*it;

for(it2=temp.begin();it2!=temp.end();it2++)

{

cout<<(\*it2).retnode()<<" "<<(\*it2).retwt()<<" ";

}

cout<<"\n";

}

return 0;

}