//minimum spanning tree

#include<bits/stdc++.h>

using namespace std;

int findminIndex(int\*weight,bool\*visited,int n){

int minIndex = -1;

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

if(!visited[i] && (minIndex == -1 || weight[i] < weight[minIndex])){

minIndex = i;

}

}

return minIndex;

}

void prims(int\*\*edges , int n){

int\*weight = new int[n];

bool\*visited = new bool[n];

int\*parent = new int[n];

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

weight[i] = INT\_MAX;

visited[i] = false;

}

parent[0] = -1;

weight[0] = 0;

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

int minIndex = findminIndex(weight,visited,n);

visited[minIndex] = true;

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

if(edges[minIndex][j] != 0 && !visited[j]){

if(edges[minIndex][j] < weight[j]){

weight[j] = edges[minIndex][j];

parent[j] = minIndex;

}

}

}

}

int ans = 0;

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

if(parent[i] < i){

ans += weight[i];

}else{

ans += weight[i];

}

}

cout<<ans;

}

int main(){

int n;

int e;

cout<<"Enter number of vertices and number of edges: ";

cin>>n>>e;

int\*\*edges = new int\*[n];

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

edges[i] = new int[n];

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

edges[i][j] = 0;

}

}

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

int f,s,weight;

cout<<"Enter city1:";

cin>>f;

cout<<"Enter city2:";

cin>>s;

cout<<"Enter weight: ";

cin>>weight;

edges[f][s] = weight;

edges[s][f] = weight;

}

cout<<endl;

prims(edges,n);

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

delete[]edges[i];

}

delete []edges;

}