# TangledInCables.cpp

```
#include <bits/stdc++.h>
using namespace std;

typedef string node;
typedef pair<double, node> edge;
typedef map<node, vector<edge> > graph;

int main(){
    double length;
    while (cin >> length){
        int cities;
        cin >> cities;
        graph g;
        for (int i=0; i<cities; ++i){
            string s;
            cin >> s;
            g[s] = vector<edge>();
        }
        int edges;
        cin >> edges;
        for (int i=0; i<edges; ++i){
            string u, v;
            double w;
            cin >> u >> v >> w;
            g[u].push_back(edge(w, v));
            g[v].push_back(edge(w, u));
        }

        double total = 0.0;
        priority_queue<edge, vector<edge>, greater<edge> > q;
        q.push(edge(0.0, g.begin()->first));
        set<node> visited;
        while (q.size()){
            node u = q.top().second;
            double w = q.top().first;
            q.pop();

            if (visited.count(u)) continue;
            visited.insert(u);
            total += w;
            vector<edge> &vecinos = g[u];
            for (int i=0; i<vecinos.size(); ++i){
                node v = vecinos[i].second;
                double w_extra = vecinos[i].first;
                if (visited.count(v) == 0){
                    q.push(edge(w_extra, v));
                }
            }
        }

        if (total > length)
            cout << "Not enough cable" << endl;
        else
            printf("Need %.1lf miles of cable\n", total);
    }
    return 0;
}
```