

```

#include <bits/stdc++.h>

using namespace std;

#include <unistd.h>

#define redirect_stdin FILE*spojectest=fopen("/tmp/spojectest.in","r");dup2(fileno(spojectest),STDIN_FILENO);

int dijkstra(int mainsrc, vector<vector<pair<int, int> > >& graph, int maindest)
{
    vector<int> route(graph.size(), numeric_limits<int>::max());
    priority_queue<pair<int, int>, vector<pair<int, int> >, greater<pair<int, int> > > pqueue;
    pqueue.push(make_pair(0, mainsrc));
    while(pqueue.size())
    {
        int weight = pqueue.top().first;
        int dest = pqueue.top().second;
        pqueue.pop();
        if(dest == maindest)
            return min(route[dest], weight);
        if(route[dest] > weight)
        {
            route[dest] = weight;
            for (int i = 0; i < graph[dest].size(); ++i)
            {
                pqueue.push(make_pair(
                    route[dest] + graph[dest][i].second,
                    graph[dest][i].first));
            }
        }
    }
    return numeric_limits<int>::max();
}

```

```
}
```

```
int main(int argc, char const *argv[])
{
    ios::sync_with_stdio(0);cin.tie(0);
    // redirect_stdin;

    int t;
    cin >> t;
    while(t--) {
        map<string, int> citydb;
        vector<vector<pair<int, int> > > graph;
        int n;
        cin >> n;
        for (int city_id = 0; city_id < n; ++city_id)
        {
            string city;
            cin >> city;
            citydb[city] = city_id;
            graph.push_back(vector<pair<int, int> >());
            int connections;
            cin >> connections;
            for(int i = 0; i < connections; i++)
            {
                int dest;
                int weight;
                cin >> dest >> weight;
                graph[city_id].push_back(make_pair(dest - 1, weight));
            }
        }
    }
}
```

```
}  
  
int queries;  
  
cin >> queries;  
  
for (int i = 0; i < queries; ++i)  
{  
    string src;  
    string dest;  
    cin >> src >> dest;  
    cout << dijkstra(citydb[src], graph, citydb[dest]) << '\n';  
}  
}  
  
return 0;  
}
```