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


unordered_map<string, vector<pair<string,int>>> owe_map;
unordered_map<string, vector<pair<string,int>>> debt_map;
void fillOweMap(vector<vector<string>> data){

    for(int i=0;i<data.size();i++){
        string s1=data[i][0];
        string s2=data[i][0];
        int value=stoi(data[i][2]);
        owe_map[s1].push_back({s2,value});
    }
}

vector<pair<string,int>> calculateOwedMoney(string name){
    cout<<name<<"owes:\n";
    vector<pair<string,int>> ans;
    for(auto it:owe_map[name]){
        cout<<it.first<<"->"<<it.second<<"\n";
       ans.push_back({it.first,it.second});
    }
    return ans;
}
pair<string,int> mostDebtPerson(){
    int total;
    priority_queue<pair<int,string>> pq;
    for(auto it:owe_map){
        int total=0;
        for(auto it2:owe_map[it.first]){
            total+=it2.second;
        }
        pq.push({total,it.first});
    }
    cout<<pq.top().second<<pq.top().first;
    return {pq.top().second, pq.top().first};
}

void moneySinglePersonOwed(string name){
    int totalOwedMoney=0;
    for(auto it:owe_map){
        for(auto it2:owe_map[it.first]){
            if(it2.first==name)
                totalOwedMoney+=it2.second;
        }
    }
    cout<<totalOwedMoney;
}

void mostOwedPerson(vector<vector<string>> data){

    for(int i=0;i<data.size();i++){
        string s1=data[i][0];
        string s2=data[i][1];
        int value=stoi(data[i][2]);
        debt_map[s2].push_back({s1,value});
    }
```

```
    int total;
    priority_queue<pair<int,string>> pq;
    for(auto it:debt_map){
        int total=0;
        //cout<<it.first<<it.second.size();
        for(auto it2:debt_map[it.first]){
            total+=it2.second;
        }
        pq.push({total,it.first});
    }
    cout<<pq.top().second<<pq.top().first;
    //return {pq.top().second, pq.top().first};

}


int main()
{
    vector<vector<string>> data;
    data.push_back({"M","N", "20"});
    data.push_back({"L","M", "70"});
    data.push_back({"N","L", "40"});
    data.push_back({"N","K", "100"});
    fillOweMap(data);
    string in;
    while(1){
        cout<<"Input the person you want to see the money owed\n\n";
        cin>>in;
        calculateOwedMoney(in);
        cout<<"\n\nEnter the money ower by the person name\n\n";
        cin>>in;
        moneySinglePersonOwed(in);

        cout<<"\n\nMaximum money Debt by: ";
        mostDebtPerson();
        cout<<"\n\nMaximum money owed by: \n\n";
        mostOwedPerson(data);
    }
    return 0;
}
```