vector - struct - Problema Imposto da Federação Galáctica

```
#include <iostream>
#include <vector>
#include <utility>
#include <algorithm>
#include <iterator>
#include <string>
using namespace std;

struct planeta{
    string nome;
    int imp;
};

bool cmp(planeta a, planeta b){
    return (a.nome < b.nome);
}

int main() {
    vector<planeta> planetas;
    planeta plan;
    int n,i,num,total=0;
    cin >> n;
    for(i=0;i<n;i++){
        cin >> plan.nome >> plan.imp;
        planetas.push_back(plan);
    }
    cin >> num;
    string nom;
    vector<planeta>::iterator it;
    for(i=0;i<num;i++){
        cin >> plan.nome;
        it = lower_bound(planetas.begin(),planetas.end(),plan,cmp);
        if (it->nome == plan.nome)
            total += it->imp;
    }
    cout << total << endl;
}
```

vector - pair - Problema Imposto da Federação Galáctica

```
#include <iostream>
#include <vector>
#include <utility>
#include <algorithm>
#include <iterator>
#include <string>
using namespace std;

bool cmp (pair<string, int> a, pair<string, int> b){
    return a.first < b.first;
}

int main() {
    vector<pair<string,int>> planetas;
    pair<string,int> plan;
    int n,i,num,total=0;
    cin >> n;
    for(i=0;i<n;i++){
        cin >> plan.first >> plan.second;
        planetas.push_back(plan);
    }
    cin >> num;
    vector<pair<string,int>>::iterator it;
    for(i=0;i<num;i++){
        cin >> plan.first;
        it = lower_bound(planetas.begin(),planetas.end(),plan,cmp);
        // int pos = it - planetas.begin();
        // if (planetas[pos].first == plan.first){
        if (it->first == plan.first){
            total += it->second;
        }
    }
    cout << total << endl;
}
```