

```

#include <fstream>
#include <string>
#include <map>
#include <sstream>
#include <iostream>
#include <vector>

using namespace std;

using tabla = map < string, int >;

void positivo(tabla &mapa, string const& a) {
    ++mapa[a];
}

void negativo(tabla &mapa, string const& a) {
    --mapa[a];
}

vector<pair<string, int>> mostrar(tabla const& mapa) {

    vector<pair<string, int>> calificados;

    for (pair<string, int> const& par : mapa) {
        if (par.second != 0) {
            calificados.push_back(par);
        }
    }

    return calificados;
}

bool resuelveCaso() {

    int n;
    string a, c;

    cin >> n;

    if (n == 0) {
        return false;
    }
    else {

        tabla listado;
        for (int i = 0; i < n; i++) {
            getline(cin, a);
            getline(cin, a);
            cin >> c;
            if (c == "CORRECTO") {

                positivo(listado, a);
            }
        }
    }
}

```

```

        }
        else {

            negativo(listado, a);
        }
    }

    vector<pair<string, int>> solucion = mostrar(listado);

    for (size_t i = 0; i < solucion.size(); i++) {
        cout << solucion[i].first << ", " << solucion[i].second << endl;
    }

    cout << "---" << endl;

    return true;
}
}

int main() {
#ifdef DOMJUDGE
    std::ifstream in("datos.txt");
    auto cinbuf = std::cin.rdbuf(in.rdbuf()); // save old buf and redirect std :: cin to
casos .txt
#endif

    while (resuelveCaso());

#ifdef DOMJUDGE // para dejar todo como estaba al principio
    std::cin.rdbuf(cinbuf);
    system(" PAUSE ");
#endif
    return 0;
}

```