#include <iostream>

#include <vector>

#include <algorithm>

**using** **namespace** std;

**struct** Result {

string city;

string district;

int voters;

int yes;

int no;

int blank;

int null;

};

void read\_data(vector<Result>& v) {

Result r;

**while** (cin >> r.city >> r.district >> r.voters >>

r.yes >> r.no >> r.blank >> r.null) {

v.push\_back(r);

}

}

bool compara(**const** Result &a, **const** Result &b) {

**return** a.district < b.district;

}

void compute\_and\_print ( **const** vector < Result >& v ) {

int size = v . size ();

**if** ( size == 0) **return** ;

*// we use a 'Result ' variable to store the city with maximum*

*// participation in current district*

Result maxcity ;

*// init the maximum with the data from first city*

maxcity = v [0];

*// Check each city ( they are sorted by district , so all*

*// cities from the same district are together )*

**for** (int i = 1; i < size ; ++ i ) {

**if** ( v [ i ]. district == v [i -1]. district ) {

*// the current city district is equal to the previous ,*

*// we check if current city it is a better maximum*

**if** ( participation ( v [ i ]) > participation ( maxcity ))

maxcity = v [ i ];

}

**else** {

*// the current district is not equal to the previous ,*

*// so , we output the maximum of the just finished district*

print\_district\_max ( maxcity );

*// ... and we reinitialize the maximum for the*

*// newly started district with the counts of first city*

*// in this district*

maxcity = v [ i ];

}

}

*// calculate and print max city of last district*

print\_district\_max ( maxcity );

}

int main() {

vector<Result> v;

read\_data(v);

sort(v.begin(), v.end(), compara);

compute\_and\_print(v);

}

**typedef** vector<vector<int> > Mat;

|  |  |
| --- | --- |
| Mat llegirMat(int n, int m) {  Mat A(n, vector<int>(m));  **for** (int i = 0; i < n; ++i) {  **for** (int j = 0; j < m; ++j) {  cin >> A[i][j];  }  }  **return** A;  } | void escriureMat(**const** Mat &A) {  **for** (int i = 0; i < A.size(); ++i) {  cout << A[i][0];  **for** (int j = 1; j < A[i].size(); ++j) cout << " " << A[i][j];  cout << endl;  }  } |