Wikipedia X15144_en

Each digital encyclopedia article has a unique identifier and belongs to a topic. Information about encyclopedia articles and queries made by users is available. Complete the program below to list the articles by topic, so that in each subject, the most popular ones appear first.
Exam score: 4 **Automatic part:** 40%

Input

The input is first formed by an integer *n* greater than zero followed by 2*n* strings that are respectively the identifiers and topics of the *n* encyclopedia articles. This information appears in lexicographical order regarding the article identifiers. After that, the list of encyclopedia identifiers (strings) that represent all user queries done appears.

Output

A list of the encyclopedia articles classified by subject. Subjects appear in lexicographical order. For each topic, the most consulted articles appear before. Ties are solved sorting article identifiers lexicographically. Take a look at the example below.

Sample input

```
10
fermat_last_theorem maths
las_meninas arts
mona_lisa arts
napoleon_bonaparte history
penicillim_discovery science
pythagoras_theorem maths
quantum_mechanics science
trafalgar_battle history
uncertainty_principle science
waterloo_battle history

pythagoras_theorem
penicillim_discovery
mona_lisa
uncertainty_principle
waterloo_battle
quantum_mechanics
napoleon_bonaparte
trafalgar_battle
las_meninas
mona_lisa
waterloo_battle
mona_lisa
penicillim_discovery
uncertainty_principle
mona_lisa
las_meninas
uncertainty_principle
trafalgar_battle
```

Sample output

```
arts 4 mona_lisa
arts 2 las_meninas
history 2 trafalgar_battle
history 2 waterloo_battle
history 1 napoleon_bonaparte
maths 1 pythagoras_theorem
maths 0 fermat_last_theorem
science 3 uncertainty_principle
science 2 penicillim_discovery
science 1 quantum_mechanics
```

Observation

Complete the following sketch. The query list can be very long. Think about efficiency.

```
struct Article {
    string identifier;
    string topic;
    int freq;
};


//pre: n > 0
//post: returns article vector formed by input channel data
vector<Article> initialize_art_vector(int n) {
    ....
    ....
}


//pre: v is non-empty
//post: write vector on output
void write_art_vector(.... v) {
    ....
    ....
}

.......
.......

int main() {
    int n;
    cin >> n;
    vector<Article> v = initialize_art_vector(n);
    ......
    ......
    write_art_vector(v);
}
```

Problem information

Author : Pro1
Generation : 2022-01-13 16:25:16