Algorithmic Trader Coding Test

02 : 59
to test end

1
2
3
4

☆ HackLand Election

There are *n* citizens voting in this year's *HackLand election*. Each voter writes the name of their chosen candidate on a ballot and places it in a ballot box.
The candidate with the highest number of votes wins the election;
if two or more candidates have the same number of votes, then the tied candidates' names are ordered alphabetically and the *last* name wins.

Complete the *electionWinner* function in your editor. It has *1* parameter: an array of strings, *votes*, describing the votes in the ballot box. This function must review these votes and return a string representing the name of the winning candidate.

Input Format

The locked stub code in your editor reads the following input from stdin and passes it to your function:
The first line contains an integer, *n*, denoting the size of the *votes* array.
Each line *i* of the *n* subsequent lines (where $0 \leq i < n$) of strings contains a citizen's vote in the form of a candidate's name.

Constraints

- $1 \leq n \leq 10^4$

Output Format

Your function must return a *string* denoting the name of the *winner*. This is printed to stdout by the locked stub code in your editor.

Sample Input 1

```
10
Alex
Michael
Harry
Dave
Michael
Victor
Harry
Alex
Mary
Mary
```

```
Michael
```

Explanation 1

votes = {"Alex", "Michael", "Harry", "Dave", "Michael", "Victor", "Harry", "Alex", "Mary", "Mary"}
Alex, Harry, Michael, and Mary are all tied for the highest number of votes. Because Michael is alphabetically last, we return his name as the winner.

Sample Input 2

```
10
Victor
Veronica
Ryan
Dave
Maria
Maria
Farah
Farah
Ryan
Veronica
```

Sample Output 2

```
Veronica
```

Explanation 2

votes = {"Victor", "Veronica", "Ryan", "Dave", "Maria", "Maria", "Farah", "Farah", "Ryan", "Veronica"}
Veronica, Ryan, Maria, and Farah are all tied for the highest number of votes. Because Veronica is alphabetically last, we return her name as the winner.

YOUR ANSWER

We recommend you take a quick tour of our editor before you proceed.
The timer will pause up to 90 seconds for the tour. Start tour

Algorithmic Trader Coding Test

02 : 59
to test end

1
2
3
4

```
8
9  /*
10  * Complete the function below.
11  */
12 char* electionWinner(int votes_size, char** votes) {
13
14
15 }
16
```

```
17 int main() {↔}
39
```

Line: 10 Col: 1

Test against custom input

Run Code

Submit code & Continue

(You can submit any number of times)

Download sample test cases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*