CS1020E | Lab 4 | Exercise 1

Social Network

Objective

The objective of this exercise is to practice OOP and learn to use the STL `vector` class.

Problem Description

In a social network, people can create, join and quit groups. We are interested in knowing:

1. **Which group** has the maximum number of members.
2. **Who** knows the maximum number of people in the social network.

We assume that people in the social network know each other if and only if they are in some common group. We say that two persons *A* and *B* know each other *via* Group *G*, if and only if both *A* and *B* are in Group *G*.

In this Lab exercise, we will implement such a system that answers the above two types of questions, as well as simulates the users' creating, joining and quitting groups.

Add your code only to the parts of the files indicated. Do not modify any other part of the given code, and do not add new files.

Inputs

The first line of the input contains an integer *N* (*N* <= 500), which is the number of operations.

Each of the *N* lines followed is in one of the following forms:

1. **`Createjoin` *A G***. Person *A* joins Group *G*. If *G* does not exist yet, then *A* creates *G* first and then joins it. *A* and *G* are names (strings). If *A* is already in *G*, the input will not have record of him/her joining *G* again, unless he/she has quit from *G*.
2. **`Quit` *A G***. Person *A* quits Group *G* that he is currently in. The input will not have record of *A* quitting *G* if *A* is not in *G*.
3. **`Query 1`**. The first type of queries. The system should output the name of the group which has the maximum number of members. If multiple names exist, the system should output the one which is lexicographically the smallest.
4. **`Query 2`**. The second type of queries. The system should output the name of a person who knows the maximum number of people in the social network. If multiple names exist, the system should output the one which is lexicographically the smallest.

We assume that: first, there is no group at the beginning; second, all group names as well as person names are **distinct**, and there is no white space in each of them. All characters in the names are '0'-'9', 'A'-'Z', or 'a'-'z'. Each name contains at most 20 characters.

Outputs

For each query in the input, output a line containing the corresponding name, which is the answer to that query.

Sample Input

```
18
Createjoin Acer Zur
Createjoin Weixiang Zur
Createjoin Shuhe Zur
Createjoin Weixiang Yuan
Createjoin Acer Yuan
Quit Weixiang Zur
Query 1
Query 2
Createjoin Zing Apple
Createjoin Weixiang Apple
Createjoin Qing Dota
Createjoin Weixiang Dota
Createjoin Zing Dota
Query 1
Createjoin Qing Yuan
Createjoin Qing Apple
Createjoin Acer Dota
Query 2
```

Sample Output

```
Yuan
Acer
Dota
Acer
```

Submission

You need to submit **ALL** your completed skeleton **`*.cpp`** and **`*.h`** files to CodeCrunch (https://codecrunch.comp.nus.edu.sg/) before the specified deadline. We will take only your latest submission.

Late submissions will not be accepted. The submission system in CodeCrunch will automatically close at the deadline.