Problem I: The Grandest Social Gathering

Filename: *social2*

Time limit: *4 seconds*

Welcome to The Grandest Social Gathering, Round Two! You're hosting another magnificent event, and your role is crucial in orchestrating it flawlessly. Your responsibilities include managing the guest list, addressing queries about attendees, and ensuring that the **custom** order of groups aligns with the spirit of the gathering.

You'll lead guests through this enchanting experience (again), offering insights into the size of specific groups and unveiling the next group's name in the custom order.

The Problem:

Given a set of operations that involve adding, querying, and removing names. Your task is to efficiently manage the guest list, grouping people by name and sorting these groups based on the size of the group.

The Input:

The input comprises multiple scenarios, with the first line containing a positive integer, $\boldsymbol{c}$ ($1 \leq \boldsymbol{c} \leq 10$), representing the number of scenarios to be processed. Each scenario is outlined by the following lines:

- The first line is an integer, $\boldsymbol{n}$, ($1 \leq \boldsymbol{n} \leq 10^5$), the number of operations to be performed.
- The next n lines detail the operations.
    - `ADD **name**`: Welcome a new guest to the gathering.
    - `QUERY **name**`: Inquire about the size of the group with the specified name. Additionally, provide information about the "next group" in the custom order, creating a sense of anticipation. If there is no other group after, print "None".
    - `REMOVE **name**`: Bid farewell to an attendee with the specified name.
    - It is guaranteed that for QUERY and REMOVE there will be at least one person with the specified name.

All names will be in between 1 and 20 alphabetic characters long and no name will be "None."

The Output:

For each `QUERY` operation, provide the count of people with the specified name (case sensitive) and name of the next group in the custom order. If there is no other group after, print "None".

Sorting Instructions:

You arrange the guest list according to the following criteria:

1. Size of group: Groups with more members take precedence over those with fewer members.
2. Lexicographical order: In case of ties, groups are sorted lexicographically with their names.

Sample Input

```
2
6
ADD Bob
ADD Joe
QUERY Bob
QUERY Joe
ADD Joe
QUERY Joe
17
ADD Alice
ADD Bob
ADD John
ADD Rob
QUERY John
ADD Alice
ADD Rob
REMOVE Alice
ADD Bob
ADD Alice
QUERY Bob
ADD Rob
ADD John
QUERY Bob
REMOVE Bob
QUERY Alice
QUERY John
```

Sample Output

```
1 Joe
1 None
2 Bob
1 Rob
2 Rob
2 John
2 John
2 Bob
```

Sample Explanation: In the first sample, there are 2 groups at first: (Bob, 1), (Joe, 1). The group following (Bob, 1) is the Joe group, and there's no group following Joe. When Joe is added, the new group ordering is (Joe, 2), (Bob, 1), thus, the group that follows Joe is the Bob group.

In the second sample, right before the first query we have groups: (Alice, 1), (Bob, 1), (John, 1), and (Rob, 1). John is followed by Rob. After the sequence of 5 ADD/REMOVE operations, the groups are now: (Alice, 2), (Bob, 2), (Rob, 2) and (John, 1). Thus, Bob is followed by Rob. After the next two adds we have: (Rob, 3), (Alice, 2), (Bob, 2), and (John, 2). Now, Bob is followed by John. When we remove Bob we get: (Rob, 3), (Alice, 2), (John, 2), and (Bob, 1). Thus, Alice is followed by John and John is followed by Bob.