

## 7047 Chat

As everyone knows, DRD has no girlfriends. But as everyone also knows, DRD's friend ATM's friend CLJ has many potential girlfriends. One evidence is CLJ's chatting record.

CLJ chats with many girls all the time. Sometimes he begins a new conversation and sometimes he ends a conversation. Sometimes he chats with the girl whose window is on the top.

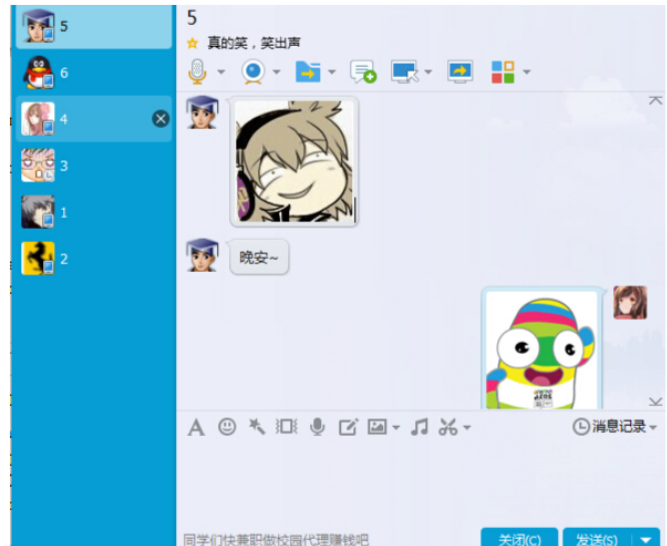
You can imagine CLJ's windows as a queue. The first girl in the queue is the top girl if no one is "always on top".

Since CLJ is so popular, he begins to assign a unique positive integer as priority for every girl. The higher priority a girl has, the more CLJ likes her. For example, GYZ has priority  $10^9$ , and JZP has priority  $10^8$  while Sister Soup has priority 1, and Face Face has priority 2.

As a famous programmer, CLJ leads a group to implement his own WM (window manager). The WM will log CLJ's operations. Now you are supposed to implement the log system. The general logging format is 'Operation #X: LOGMSG.', where X is the number of the operation and LOGMSG is the logging message.

There are several kinds of operations CLJ may use:

1. **Add**  $u$ : CLJ opens a new window whose priority is  $u$ , and the new window will be the last window in the window queue. This operation will always be successful except the only case in which there is already a window with priority  $u$ . If it is successful, LOGMSG will be 'success'. Otherwise LOGMSG will be 'same priority'.
2. **Close**  $u$ : CLJ closes a window whose priority is  $u$ . If there exists such a window, the operation will be successful and LOGMSG will be 'close  $u$  with  $c$ ', where  $u$  is the priority and  $c$  is the number of words CLJ has spoken to this window. Otherwise, LOGMSG will be 'invalid priority'. Note that ANY window can be closed.
3. **Chat**  $w$ : CLJ chats with the top window, and he speaks  $w$  words. The top window is the first window in the queue, or the "always on top" window (as described below) instead if there exists. If no window is in the queue, LOGMSG will be 'empty', otherwise the operation can be successful and LOGMSG will be 'success'.
4. **Rotate**  $x$ : CLJ performs one or more Alt-Tabs to move the  $x$ -th window to the first one in the queue. For example, if there are 4 windows in the queue, whose priorities are 1, 3, 5, 7 respectively and CLJ performs 'Rotate 3', then the window's priorities in the queue will become 5, 1, 3, 7. Note that if CLJ wants to move the first window to the head, this operation is still considered "successful". If  $x$  is out of range (smaller than 1 or larger than the size of the queue), LOGMSG will be 'out of range'. Otherwise LOGMSG should be 'success'.
5. **Prior**: CLJ finds out the girl with the maximum priority and then moves the window to the head of the queue. Note that if the girl with the maximum priority is already the first window,



this operation is considered successful as well. If the window queue is empty, this operation will fail and *LOGMSG* must be ‘empty’. If it is successful, *LOGMSG* must be ‘success’.

6. **Choose  $u$ :** CLJ chooses the girl with priority  $u$  and moves the window to the head of the queue. This operation is considered successful if and only if the window with priority  $u$  exists. *LOGMSG* for the successful cases should be ‘success’ and for the other cases should be ‘invalid priority’.
7. **Top  $u$ :** CLJ makes the window of the girl with priority  $u$  always on top. Always on top is a special state, which means whoever the first girl in the queue is, the top one must be  $u$  if  $u$  is always on top. As you can see, two girls cannot be always on top at the same time, so if one girl is always on top while CLJ wants another always on top, the first will be not always on top any more, except the two girls are the same one. Anyone can be always on top. *LOGMSG* is the same as that of the Choose operation.
8. **Untop:** CLJ cancels the “always on top” state of the girl who is always on top. That is, the girl who is always on top now is not in this special state any more. This operation will fail unless there is one girl always on top. If it fails, *LOGMSG* should be ‘no such person’, otherwise should be ‘success’.

As a gentleman, CLJ will say goodbye to every active window he has ever spoken to at last, “active” here means the window has not been closed so far. The logging format is ‘Bye  $u$ :  $c$ ’ where  $u$  is the priority and  $c$  is the number of words he has ever spoken to this window. He will always say good bye to the current top girl if he has spoken to her before he closes it.

## Input

The first line contains an integer  $T$  ( $T \leq 10$ ), denoting the number of the test cases.

For each test case, the first line contains an integer  $n$  ( $0 < n \leq 5000$ ), representing the number of operations. Then follow  $n$  operations, one in a line. All the parameters are positive integers below  $10^9$ .

## Output

Output all the logging contents.

**Hint:** This problem description does not relate to any real person in THU.

## Sample Input

```
1
18
Prior
Add 1
Chat 1
Add 2
Chat 2
Top 2
Chat 3
Untop
Chat 4
Choose 2
Chat 5
Rotate 2
Chat 4
```

Close 2  
Add 3  
Prior  
Chat 2  
Close 1

### Sample Output

Operation #1: empty.  
Operation #2: success.  
Operation #3: success.  
Operation #4: success.  
Operation #5: success.  
Operation #6: success.  
Operation #7: success.  
Operation #8: success.  
Operation #9: success.  
Operation #10: success.  
Operation #11: success.  
Operation #12: success.  
Operation #13: success.  
Operation #14: close 2 with 8.  
Operation #15: success.  
Operation #16: success.  
Operation #17: success.  
Operation #18: close 1 with 11.  
Bye 3: 2