

Crossed Book Problem

In an orderly market, the highest price a bidder is willing to pay (the best bid) for a security should be strictly lower than the lowest price a seller is offering (the best ask). However, for various reasons, markets may be locked - where the best bid is the same as the best ask - or crossed - where the best bid is higher than the best ask.

These conditions can be detected by first building a book per security. A book is essentially 2 sorted lists of orders - a list of bids, sorted with the highest bid at the top, and a list of asks, sorted with the lowest ask at the top. If the highest bid is equal to the lowest ask, the book is locked, and if the highest bid is larger than the lowest ask, the book is crossed.

Since a crossed or locked book can represent a problem with the data you are receiving, or a problem with your program processing that data, your task is to write a program that will detect and report these problems as they occur.

Input Format

There are 2 types of input messages. The first message is an “Add Order” message, which describes a new order on the book. The “Add Order” message has the following format:

timestamp A side symbol size price order-id

The second type of message is a “Modify Order” message, which changes the size and/or price of an existing order. This message has the following format:

timestamp M order-id size price

If a “Modify Order” message has a size of 0, the order is removed from the book and no further “Modify Order” messages will be sent with that order id. Input messages are guaranteed to have non-decreasing timestamps, and no two add order messages will share an order id.

Output Format

There are also two types of output messages. The first message is a “Crossed/Locked” message that should be printed out whenever a symbol becomes crossed or locked. This message has the following format:

timestamp symbol {crossed | locked} best-bid best-ask

The second message is a “Normal” message that should be printed out when a symbol that was previously crossed or locked becomes uncrossed or unlocked. This message has the following format:

timestamp symbol normal

Note that once a symbol is in a crossed or locked state, no messages should be printed for that symbol until it returns to a normal state.

Sample input

```
36000000 A B QQQQ 30 20.0000 300
36000001 A B QQQQ 20 19.9900 301
36000002 A B QQQQ 42 19.9900 302
36000003 A A QQQQ 25 20.0100 400
36000004 A A QQQQ 25 20.0200 401
36000005 M 401 25 20.0300
36000006 A A QQQQ 25 20.0000 402
36000007 A B GOOG 10 250.2300 303
36000008 A A GOOG 15 250.2200 403
36000010 M 402 0 20.0000
```

Sample output

```
36000006 QQQQ locked 20.0000 20.0000
36000008 GOOG crossed 250.2300 250.2200
36000010 QQQQ normal
```

Data Format

Field	Description
timestamp	an integer representing the time in milliseconds since midnight
side	either 'B' or 'A' depending on whether this order is a bid or an ask
symbol	an alphanumeric string up to 8 characters long that identifies a security
size	an integer no greater than 2 billion
price	a floating point number with up to 4 decimal places of precision
order-id	an integer no greater than 2 billion