Basic order book management

About

Thank you for your interest. This task is a test for candidates on Software Engineer position at Bookmap. The description below is sufficient to complete the task without prior knowledge on its topic and without external material. If you feel that there are multiple correct solution, decide which is better or send us a question (see email below). You are also welcome to ask any other question.

Please submit your solution in Java or JavaScript depending on which position you apply to. Email us either source code or a URL of a GitHub page to: hr@bookmap.com

Order book

Order Book is a collection of buy and sell orders managed by exchanges (e.g. Nasdaq) for a specific asset or financial instrument (e.g. AAPL -- Apple stock), organized by price levels. Order are sent to the exchange by either human or electronic traders and have the following properties:

Side	It can be either buy or sell
Price	Order cannot be executed/matched at a worse price
Size	Positive number of units this order aims to buy or sell

These orders are resting in the order book because their price prevents their execution/matching against orders with the opposite side. The highest price of all buy orders is called Best Bid. The lowest price of all sell orders is called Best Ask. Price levels equal or below Best Bid are called Bid prices. Price levels equal or above Best Ask are called Ask prices. Price levels between Best Bid and Best Ask are called Spread. Each price level may contain zero or more orders. The size at each price level is the sum of sizes of all orders at that price level. Here is an example of order book (B = Bid, S = Spread, A = Ask):

PRICE	SIZE	TYPE	COMMENT
100	50	A	
99	0	A	An empty Ask price
98	30	A	Best Ask: lowest non-empty ask price
97	0	S	
96	0	S	
95	40	B	Best Bid: highest non-empty bid price
94	30	B	
93	15	B	
92	77	B	

Task

Your program will read an input text file where each line is either an instruction or a query and ends with a newline character. Process each instruction according to the Action column and respond to each query by printing out corresponding output. Assume that price and size are integers.

Here is a formal description of input file format:

Line in the input file	Action
Order book updates	
u,<price>,<size>,bid	set bid size at <price> to <size>
u,<price>,<size>,ask	set ask size at <price> to <size>
Incoming Orders (*)	
o,buy,<size>	remove <size> units out of asks, cheapest ones
o,sell,<size>	remove <size> units out of bids, most expensive ones
Queries	
q,best_bid	print best bid price and size. Output example: *95,40*
q,best_ask	print best ask price and size. Output example: *98,30*
q,size,<price>	print size at specified price. Output example: *77*

(*) These orders are usually called "market orders". In case of a buy order this is similar to going to a marketplace and buying <size> units at the cheapest price available, thus removing <size> units from the ask part of the order book. Similar logic is applicable for sell orders.

Queries and instructions may appear in the file in an arbitrary order.
Assume that all lines in the input file have correct format. However, in case of a logically invalid input or query, print an error and then exit if the error is critical, i.e. it breaks the integrity of the order book.

An example of input file:

```
u,9,1,bid
u,11,5,ask
q,best_bid
u,10,2,bid
q,best_bid
o,sell,1
q,size,10
```

Corresponding output:

```
9,1
```

```
10,2
1
```

Notes

- Performance matters.
- Do not use dependencies on external libraries
- Please include in your email the link to the job posting and this document as an attachment