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. 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:

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

Execution between orders happens when prices of two orders of opposite side match, i.e. price of the buy order is greater or equal than price of the sell order. This leads to mutual annihilation of the minimal size between the two orders. For synchronization reasons orders are processed one by one as they arrive. Therefore execution is always between a resting order and the new incoming order. Orders are resting in the order book as long as their price prevents execution against orders with the opposite side. Here is an example of order book:

Price	Buy or Sell	Sizes of orders
101	S	300
100		
99	S	50,10,15
98		
97		
96	B	20,10
95	B	40

Orders processor algorithm (OPA)

Traders can place new orders. OPA describes how an order must be processed:

- As long as order's price permits, execute the order against resting orders of the opposite side according to price priority, i.e. starting from the best available price.
- Place the remaining part (if any) of the order into the order book

Each trader can request to cancel one of his resting orders.

Task

Your program will read an input text file where each line is either an action or a query and ends with a newline character. Process each action according to OPA and respond to each query by printing out corresponding output. Assume that price, size and id are integers, and side is a char.
Order ids are between 0 and 1000000 (10^6).
It is guaranteed that difference between maximum and minimum prices in the input file is not exceeding 10000 (10^4) and absolute values of prices will not exceed 1000000000 (10^9).
It is guaranteed that for every query (see below) total size will not exceed 1000000000 (10^9).
Here is a formal description of input file format:

Line in the input file	Description
Actions	
o,<id>,<side>,<price>,<size>	A new order where side is either 'b' or 's'
c,<id>	A request to cancel the order with corresponding id
Queries	
q,buyers	Print the highest price of the order book that contains buy order(s), and total size available at this price. Output example: *96,30* If there is no buy orders, print "empty" (without quotes).
q,sellers	Print the lowest price of the order book that contains sell order(s), and total size available at this price. Output example: *99,75* If there is no sell orders, print "empty" (without quotes)
q,size,<price>	Print total size at specified price. Output example: *40*

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:
o,0,b,95,40
o,1,b,96,20

q,buyers
o,2,b,96,10
o,3,s,101,300
o,4,s,99,50
o,5,s,99,10
q,sellers
o,6,s,99,15
c,1
o,7,s,91,30
q,size,95

Corresponding output is:
96,20
99,60
20

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