SETSmm Iceberg Order Exercise

The object of this exercise is to implement a simulator for SETSmm Iceberg order functionality in Java as specified in the document “SETSmm and Iceberg Orders Service & Technical Description”.

Requirements:

- The simulator will deal with only a single order book for each run.
- It will receive orders from `stdin` in a simplified ASCII format specified later in this document, which will either be limit or iceberg orders (ignore the message specifications in the technical document)
- On receipt of an order from `stdin`, the solution should apply that order to the book, possibly generating trade messages to `stdout` when there is matching to be done, and then display the current book in formats specified later
- Error handling can be kept to a minimum and error conditions such as badly formatted input will not be evaluated. However, any error messages you may wish to provide should be directed to `stderr`.

Input Format

For simplicity, this simulation only deals with order entry of either limit or iceberg orders. Orders cannot be modified or cancelled once entered into the book. Hence, the only messages to be received from `stdin` are order entry messages.

Input is received from `stdin` in the form of individual lines delimited by newlines. A line may consist entirely of whitespace characters (*empty*); begin with whitespace followed by the # character and subsequently any other characters (*comment*); in comma-separated ASCII format (*data*). Lines formatted as *empty* or *comment* shall be ignored. You may assume that lines formatted as *data* will always be correctly formatted, with no deviation from the specified types and no overflow (for simplicity, all prices are in whole pence, no fractions are allowed). *Data* lines may be one of:

Insert Limit Order

This message is contained within a single line, with comma-separated values:

Field Index	Type	Description
0	`char`	‘`B`’ for a buy order, ‘`S`’ for a sell order
1	`int`	unique order identifier
2	`short`	price in pence (> 0)
3	`int`	quantity of order (> 0)

Example:

```
B,100322,5103,7500
```

Description:

Limit order ID 100322: Buy 7,500 at 5,103p.

Insert Iceberg Order

This message is practically identical to an insert limit order message, except that it contains one extra field specifying the peak size. It is again contained within a single line, with comma-separated values. Unlike the associated technical document, for the purposes of this simulation iceberg orders only have a single order identifier (no hidden identifier) that will be used in trade messages.

Field Index	Type	Description
0	`char`	‘`B`’ for a buy order, ‘`S`’ for a sell order
1	`int`	unique order identifier (> 0)
2	`short`	price in pence (> 0)
3	`int`	quantity of order (> 0)
4	`int`	peak size (> 0)

Example:

```
S,100345,5103,100000,10000
```

Description:

Iceberg order ID 100345: Sell 100,000 at 5103p, with a peak size of 10,000.

You may assume that the peak size is never greater than the total quantity.

Output Format

All non-error output should be directed to `stdout`. For each order insert message, the solution should apply the order to the current book and generate any matched trades in order of their first matches, and then output the current book.

Trade Format

A trade message should be sent to `stdout` for each matched trade. Note that in the technical specification it states that only a single trade message will be sent for each iceberg order, even if the match occurs on more than one peak (e.g. a match size of 16,000 on an iceberg order with a peak size of 10,000 should trade once for a quantity of 16,000, not one for 10,000 and another for 6,000).

The trade message should be contained within a single line, with nothing else present on the same line, and be in comma-separated format with the following fields.

Field Index	Type	Description
0	`int`	buy order ID matched
1	`int`	sell order ID matched
2	`short`	price in pence
3	`int`	quantity

Example:

```
100322,100345,5103,7500
```

Book format

Following the receipt of an order message, and after resolving any matches in the book and outputting any generated trade messages, the solution should display the current full order book in the following example format:

- Id columns width (excluding formatting marks) = 10
- Volume columns width (excluding formatting marks) = 13
- Price columns width (excluding formatting marks) = 7
- Total width including 7 formatting marks = 67
- All headers left-justified with single space between formatting mark and start of text
- All data fields right-justified with no space between data and formatting mark
- Order IDs should be formatted as plain numbers (i.e. 103749 not 103,749), but volumes and prices should be formatted with commas (i.e. 100,000 not 100000).

```
+-----------------------------------------------------------------+
| BUY                            | SELL                           |
| Id       | Volume      | Price | Price | Volume      | Id       |
+----------+-------------+-------+-------+-------------+----------+
|1234567890|1,234,567,890| 32,503| 32,504|1,234,567,890|1234567891|
|      1138|        7,500| 31,502| 32,505|        7,777|      6808|
|          |             |       | 32,507|        3,000|     42100|
+-----------------------------------------------------------------+
```

Note 1: the volume displayed for iceberg orders should be the current peak volume, not total volume.

Note 2: the order book entries must be displayed in priority order.