Dear applicant,

As part of our selection procedure we would like to ask you to take on a small programming challenge so that we can get a good initial view on your skills, methodologies and style of programming.

Please note the following when implementing the functions.

- Please provide an implementation in C++; third-party libraries like boost are allowed but shouldn't really be necessary.
- While performance is important, we don't expect you to shave off the last milli-/micro-seconds; please do not spend time on inner-loop optimizations.
- The test should not take more than a few hours, but we give you three days to return it to us. In this way you can complete it at a time of your convenience and revisit it with a fresh pair of eyes, if you wish to do so.
- Do not worry about formatting the output, we care more about correctness.

Should you have any questions about this challenge or otherwise, please don't hesitate to contact us!

The Challenge

Given a simple, text-based, data file with a captured order-by-order feed for a single day from some exchange, containing records of the following format (with newline-characters delimiting the records):

```
timestamp;symbol;order-id;operation;side;volume;price
```

For example:

```
14:17:21.877391;DVAM1;00000001;I;BUY;100;12.5
14:17:22.123523;DVAM1;00000002;I;SELL;37;13.5
14:17:22.343883;DVAM1;00000001;A;BUY;100;12.7
14:17:24.737292;DVAM1;00000003;I;SELL;37;13.3
14:17:24.893811;DVAM1;00000004;I;BUY;55;12.7
14:17:25.883711;DVAM1;00000002;C;SELL;37;13.5
```

Where 'operation' can be one of:

`I` = `Insert / new order`
This is a new order added to the book; it will have a new/unique order-id.

`C` = `Cancel / delete order`
The order with the given order-id is to be removed from the book.

`A` = `Amend / modify order`
The order with the given order-id is to be changed to the new volume and/or price.

Orders, once inserted, join an order book. This order book is used to match up offers to buy and sell; if there is a match, the orders are said to be filled and a trade occurs. In this simple assignment however, no trades occur, so an order stays active until it is cancelled.

Can you please implement the following three functions and run them against the accompanying `orders.dat` file?

1. **`OrderCounts()`**
   - Counts the total number of orders per symbol.

2. **`BiggestBuyOrders(symbol="DVAM1")`**
   - Finds the top 3 biggest BUY orders in terms of volume for a specific symbol.

3. **`BestSellAtTime(symbol="DVAM1", timestamp="15:30:00")`**
   - Finds the best SELL price and related order volume for a specific symbol at a given time.

Make sure you re-read the assignment carefully, and think about how to interpret the records in the file for each of these tasks.

Please send your results and code (in a zip file) to recruitment@davinciderivatives.com

Good luck and thank you for your time!
Da Vinci Derivatives