Order Book Programming Exercise

Produce a program which maintains price-time limit order books, one per trading symbol, accepting orders and cancels from a file and publishes top of book (best bid and ask) changes for each book. Trades or matching are optional, see details below.

Requirements

Input

Read a file of order and cancel messages on an input thread.

See provided input file inputFile.csv for format, supporting three types of transactions: new or modify order ‘O' and cancel order ‘C’.

Input:

- N = new order
- C = cancel
- F = flush

Order Book Processing

Order book is price-time for bids and asks, and the orders join their respective book sides (bid and ask) in price then time priority

Orders that cross the book should be rejected.

Bonus: Enable matching and verify scenarios 13 and 14 (this step is optional), orders that cross the book should be traded.

Output

Publish on output thread to console/stdout.

- publish order or cancel acknowledgement format:
  - **A, userId, userOrderId**
- publish changes in Top Of Book per side using format, use ‘-‘ for side elimination:
  - **B, side (B or S), price, totalQuantity**
- publish rejects for orders that would make or book crossed:
  - **R, userId, userOrderId**
- publish trades (matched orders) format:
  - **T, userIdBuy, userOrderIdBuy, userIdSell, userOrderIdSell, price, quantity**

Bonus: create unit tests around the book interface, shortcut: convert input scenarios to unit tests, provide more scenarios time permitting.

Test Outputs

Outputs are provided for odd scenarios in outputFile.csv, its expected that you generate your own expected outputs for even numbered scenarios and validate your output against it (strip comments and blank lines out).

Project

Use make at a minimum, use gcc/g++ 7.0+.

Tar, gzip and submit. Please do not include shared libraries, object files or executables.

Provide a readme.txt file describing how to build, run.

Bonus: provide documentation, project structure, architectural aspects, threads, classes etc. Include improvements you would make if you had more time. **Bonus**: containerize, provide instructions with a dockerFile to build and run via docker.

Important Note: Please do not submit any code that is derived from proprietary code/or code you worked on for another company previously.

Hint: if you operate faster in Visual Studio or some-other GUI C++ IDE, feel free to develop the program there, then transfer and build your project in linux.