1 PROBLEM

You are required to implement a matching engine as part of the exchange simulator in a back testing platform.

2 BACKGROUND

Order matching engines are one of the core components in modern exchanges. This component accepts buy and sell orders from clients and matches them up according to various algorithms. One algorithm adopted by most exchanges is price/time priority, which means orders with better price would get matched first and orders at the same price are matched according to the sequence they enter the matching engine.

3 REQUIREMENTS

- The matching engine accepts a stream of orders. Each order has unique id, symbol, side, price, style, size and client name.
- Two basic types of transactions should be supported: insert and cancel.
- Both market and limit order should be supported.
- When order is filled/partial filled, the matching engine will send a trade message to the client. Each trade message should only have one trade price and trade quantity.
- An interface for configurable matching logic is required, for example a client's own buy/sell orders may not cross with each other.

Instructions for the Coding Exercise

- The program must be implemented in C++ 11/14/17
- Boost or C++ standard libraries can be used
- GCC or Visual Studio can be used to build your program
- Your solution should be written for a real-time system and performance should be considered
- Submitted code should be robust and reasonably crash-resistant where necessary
- If you find any areas of the problem ambiguous choose your interpretation of the problem but document your assumptions
- Zip the solution and email us back by the stipulated date