Part II includes title page, system design, system features and user documents.

Title Page:

Solo: This work has been done by Zhen Qian (Martin).

System Design

Part I: class declaration

(1) **struct order function**: linked five attributes together like real order format, used in order operations.

(2) **compareBid class function**: give the highest bid order price priority with time stamp screening.

(3) **compareAsk class function**: give the lowest ask order price priority with time stamp screening.

Part II: simulate stock order book with the orders file.

(1) Input orders file, named as "**orders.txt**".

(2) Build empty **bid orders book** and **ask order book** with priority_queue container, set up empty **match book** using stack. Priority_queue will give the price priority, thus top element of bid book has the highest order price and top element of ask book has the lowest order price. If there is match results, push them in stack container, which is last in first out (LIFO).

(3) Deal with the orders. If the bid order price is great than or equals to ask price, check the order quantity, the smaller quantity order will be disappeared in order book and record match results in match book. If bid quantity equals ask quantity, both are disappeared in order books and record both in match book.

Part III: simulate stock order book with screen input.

(1) Design **a menu system** for order book simulation

(2) User can query current order match situation by input the order file

(3) User can query current order book feature, such as bid/ask order price range, top bid/ask orders price/shares, or top 3 orders price/shares

(4) User can input a new order, cancel orders or modify orders.

System Features

(1) Users can run orders from order file, like the server in stock exchange. **(Extra feature)**

(2) Users can query order match results via menu system. **(Extra feature)**

(3) Users can query current bid/ask order' highest price and lowest price via menu system.

(4) Users can query top bid/ask orders situation via menu system, show order time/order ticker/ order type/order quantity/order price.

(5) Users can query top 3 bid/ask orders via menu system, show important information about order price/ order shares.

(6) Users can input a new order via menu system. After input a new order, the program will run bid/ask match function and show the results.

(7) Users can modify current bid/ask orders.

(8) Users can cancel current bid/ask orders.

Argument for extra 10 points:

Running the order file is a fantastic feature that mimics the real computer server in stock exchange. Every second the program can deal with thousands of bid/ask orders and record the results in match book for further query. Users can use match book to record any bid/ask match results, which is very important information.

Users Manual

(1) Users can write bid/ask orders in file, using any file name to substitute the “orders.txt” file.

(2) User can enter 0 for exit program.

(3) User can enter 1 looking at the filled/part filled orders via menu system.

(4) User can enter 2 for view bid price order range via menu system.

(5) User can enter 3 for view ask price order range via menu system.

(6) User can enter 4 for current top order (Time, Ticker, Type, Quantity, Price) via menu system.

(7) User can enter 5 for current top three orders (Price, Quantity) via menu system

(8) User can enter 6 for input a new order (Time, Ticker, Type, Quantity, Price) and view bid/ask match result.

(9) User can enter 7 for modify order (Type, Quantity, Price).

(10) Users can enter 8 for cancel order (Type, Price).