宽德投资 订单簿项目 陈诗健

## 一、【项目概述】

订单簿遵循价格优先、时间优先的原则。在订单簿中，会显示买卖方向相同、同一价格下不同订单的数量总和。

限价单：一个新进限价单，会和对应订单簿(如买簿)中的最优价进行交易撮合。若该限价单没有被完全交易，则会被添加进另一订单簿(即卖簿)的合适位置。

市价单：查询资料知道，市价委托有多种。如“最优五档即时成交剩余撤销”、“最优五档即时成交剩余转限价”等。我采用的是“最优五档即时成交剩余撤销”。

撤单：下单过程中，若该订单被添加进订单簿。则返回一个挂单回执，内容是一个挂单编号。撤单时，输入该编号，即可完成操作。

成交价格：两订单进行撮合，按照时间早的订单价格进行交易。

【输入】

Input.txt :

**限价单**

| User_Id | Order_Action | Order_Type | Order_Direction | Price | Volumn | Order_Id |
|---|---|---|---|---|---|---|
| 1001 | A | L | B/A | 99.9 | 1000 | 1 |

**市价单**

| User_Id | Order_Action | Order_Type | Order_Direction | Volumn |
|---|---|---|---|---|
| 1002 | A | M | B/A | 1000 |

**撤单**

| User_Id | Order_Action | Order_Id |
|---|---|---|
| 1003 | R | 2 |

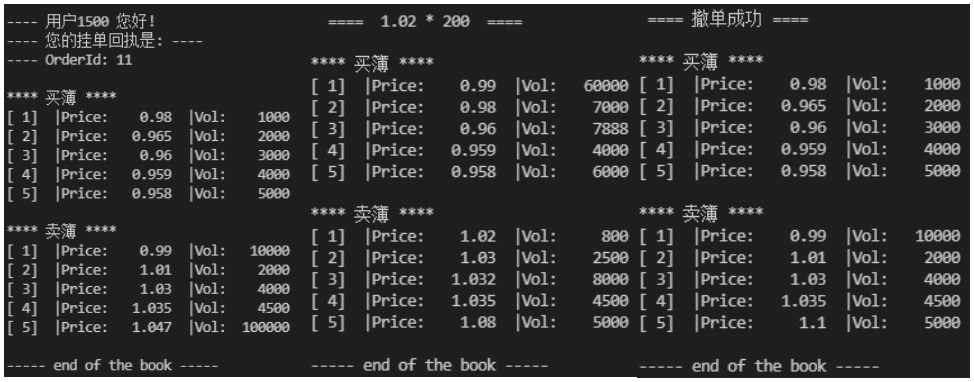
**Order_Action:** A(Add:下单) R(Reduce:撤单)

**Order_Type:** L(Limit:限价) M(Market:市价)

**Order_Direction:** B(Bid:买入) A(Ask:卖出)

【输出】:

**限价单** **市价单** **撤单**




【①限价单只要进簿了，都会返回一个挂单回执；②市价单只显示了交易信息，即“====1.02*200 ====”处，应该添加更多信息，如是否最优五档即时成交成功等；③有一个地方还要大改：每一笔交易都应该返回给对应用户一个回执，告知用户你的某个订单被交易了多少/完全交易】【result.log 写入日志信息】

## 二、【设计架构】

此项目共设计了四个类：

**Logger 类：**打印日志信息

**Order 类：**用于订单信息的存储

**Book 类：**用于买卖订单簿的构造、信息存储、限价单的交易撮合、市价单的交易撮合、撤单操作、信息打印

**Engine 类：**根据订单信息调用 Book 类中不同的方法，完成交易撮合。

**【CreateOrder.h/CreateOrder.cpp：读取文件，分割字符，存储新订单】**

```
class Engine
{
    public:
        void Update(Order& New_Order,int Level);
    private:
        Book Sell_Book;     // 卖簿
        Book Buy_Book;      // 买簿
        std::map<long long,Order> Id_Order;   // Order_ID —> Order 用于撤单，由id可以得到该Order的Price,这样就可以进入Book类中的Order_List了
        long long num;
};
```

其中 Book 类是最为关键的

**Book 类：**

**【存储字段】:**

①Order_List: 类型：map<double,map<long long,Order>>

映射关系为：Price(价格)——>Nano_Time(时间)——>Order(订单)

用于维护每一个簿中的订单信息；实现满足价格优先、时间优先的基本准则

计算机处理太快了，如果只精确到秒，会产生同一时间映射多个订单的情况

②Price_TotalVolumn: 类型：map<double, long long>

映射关系：Price(价格)——>总订单量

主要用于打印同一价格下的订单总数量

③Id_Order: 类型：map<long long,Order>

映射关系：Order_Id(订单编号)——>Order(订单)

用于实现撤单操作

**【方法】:**

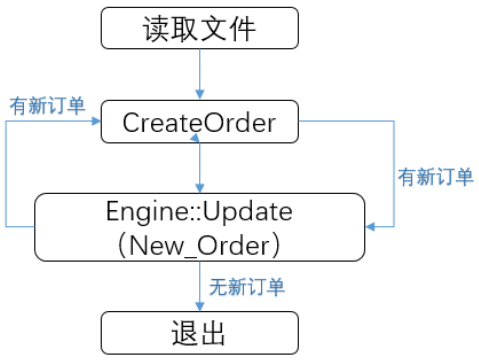
限价单交易：void Order_Deal(Order& New_Order);

市价单交易：void Order_Market_Deal(Order& New_Order);

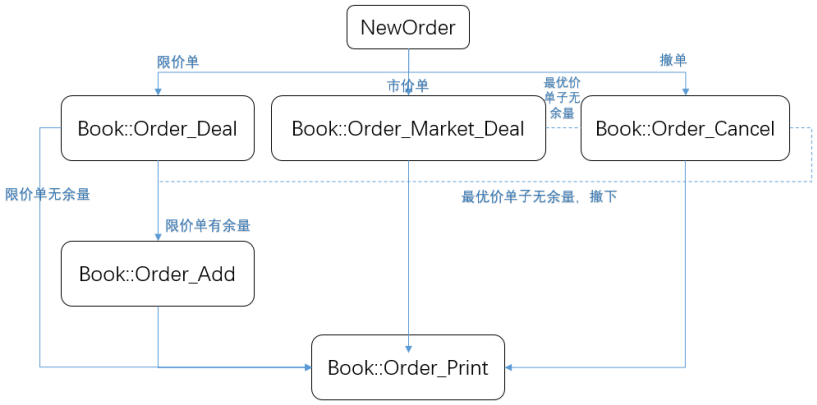
若订单没被完全交易，则加入订单簿：void Order_Add(Order& Left_Order);

撤单操作：void Order_Cancel(long long Order_Id);

打印订单簿信息：void Order_Print(char Book_Type);



先对输入文件进行字符处理，不断构建新的 Order，然后根据 Order 不同的信息，进行处理。Engine 类中有两个 Book 对象，Sell_Book 和 Buy_Book，比如一个 Order 是限价买单，则该 Order 先与 Sell_Book 中的 Order_List 进行交易撮合，即 Order_Deal。在撮合过程中，Sell_Book 中的卖单可能会被完全成交，则记录该卖单的 Order_Id，最后对这些 Order_Id 实现 Order_Cancel。另外，如果这个限价买单没有被完全成交，则调用 Buy_Book 的 Order_Add。在这整个过程中，Buy_Book 和 Sell_Book 的三个存储字段都是不断更新以维护订单信息的。市价单同理，不过市价单是肯定不会加入到订单簿中的。若前 5 档无法即时成交，剩余的会被撤销。在撮合过程中，卖/买簿被完全成交的单子也要调用 Order_Book.




【Order_Deal 中买单的交易过程】

```
case 'B':     // 限价买单
{
    for(auto iter=this->Order_List.begin();iter!=this->Order_List.end();iter++)   // 根据价格由低到高排序
    {
        if(flag==1) break;     // 若买单已经完全成交
        if(New_Order.Price<iter->first) break;    // 若买单小于卖簿的最优价，即无法成交，则退出
        for(auto iter1=this->Order_List[iter->first].begin();iter1!=this->Order_List[iter->first].end();iter1++)  // 同一价格根据时间从先到后排序
        {
            if(New_Order.Volumn<iter1->second.Volumn)   // 若买单可以被完全成交
            {
                Deal_Print(iter->first,New_Order.Volumn);
                iter1->second.Volumn-=New_Order.Volumn;   // 更新卖簿订单信息
                this->Price_TotalVolumn[iter->first]-=New_Order.Volumn;   //维护卖簿的Price_TotalVolumn
                this->Id_Order[iter1->second.Order_Id].Volumn-=New_Order.Volumn;   //维护卖簿的Id_Order
                New_Order.Volumn=0;
                flag=1;
                break;
            }
            // 若卖簿最优价被完全成交，买单有余量
            Deal_Print(iter->first,iter1->second.Volumn);
            this->Price_TotalVolumn[iter->first]-=iter1->second.Volumn;   //维护卖簿的Price_TotalVolumn
            this->Id_Order[iter1->second.Order_Id].Volumn-=iter1->second.Volumn;  //维护卖簿的Id_Order
            New_Order.Volumn-=iter1->second.Volumn;  // 更新买单信息
            temp.push_back(iter1->second.Order_Id);      //记录卖簿Order_Id
        }
    }
    for(auto iter=temp.begin();iter!=temp.end();iter++)
        this->Order_Cancel((*iter));     // 将被完全成交的卖簿订单撤下
    break;
```

（第一层循环实现价格优先，第二层循环是实现同一价格下时间优先。）

【Order_Cancel】

```
void Book::Order_Cancel(long long Order_Id)
{
    this->Order_List[this->Id_Order[Order_Id].Price].erase(this->Id_Order[Order_Id].Nano_Time);  // 更新买/卖簿Order_List信息
    if(this->Price_TotalVolumn[this->Id_Order[Order_Id].Price]==0)  // 如果同一价格下的订单数已为0
    {
        this->Order_List.erase(this->Id_Order[Order_Id].Price);
    }
}
```

撤单操作中指定的订单，和“原本在订单簿中但被完全交易的订单”，都会进行

Order_Cancel。如果同一价格下已经没有订单量了，我会将 Order_List 中此价格下的映射都删除掉。

## 三、【具体代码运行演示】

执行： ./OrderBook_Engine

```
chenshijian@wizardquant-ProLiant-DL380-Gen9:~/OrderBook_Engine$ cmake .
-- Configuring done
-- Generating done
-- Build files have been written to: /home/chenshijian/OrderBook_Engine
chenshijian@wizardquant-ProLiant-DL380-Gen9:~/OrderBook_Engine$ make
Scanning dependencies of target OrderBook_Engine
[ 14%] Building CXX object CMakeFiles/OrderBook_Engine.dir/main.cpp.o
[ 28%] Building CXX object CMakeFiles/OrderBook_Engine.dir/Logger.cpp.o
[ 42%] Building CXX object CMakeFiles/OrderBook_Engine.dir/Order.cpp.o
[ 57%] Building CXX object CMakeFiles/OrderBook_Engine.dir/Book.cpp.o
[ 71%] Building CXX object CMakeFiles/OrderBook_Engine.dir/Engine.cpp.o
[ 85%] Building CXX object CMakeFiles/OrderBook_Engine.dir/CreateOrder.cpp.o
[100%] Linking CXX executable OrderBook_Engine
[100%] Built target OrderBook_Engine
chenshijian@wizardquant-ProLiant-DL380-Gen9:~/OrderBook_Engine$ ./OrderBook_Engine
[WELCOME] /home/chenshijian/OrderBook_Engine/Logger.cpp 2022-02-16 10:39:33 : === Start logging ===
```

执行结果示例：输入解释性文件参照 Input_解释文档.txt；如果需要调试 10 档，则将解释文档中 5 档初始化输入部分替换成 10 档初始化输入部分即可。

**1、订单簿(5 档)初始化**

```
1001 A L B 0.98 1000
1002 A L B 0.957 6000
1001 A L B 0.959 4000
1003 A L A 0.99 10000
1005 A L B 0.958 5000
1004 A L A 1.047 100000
1006 A L B 0.96 3000
1005 A L A 1.035 4500
1009 A L A 1.1 5000
1333 A L A 1.01 2000
1450 A L A 1.03 4000
1500 A L B 0.965 2000
```

```
---- 用户1500 您好!
---- 您的挂单回执是: ----
---- OrderId: 11

**** 买簿 ****
[ 1]  |Price:    0.98  |Vol:    1000
[ 2]  |Price:   0.965  |Vol:    2000
[ 3]  |Price:    0.96  |Vol:    3000
[ 4]  |Price:   0.959  |Vol:    4000
[ 5]  |Price:   0.958  |Vol:    5000

**** 卖簿 ****
[ 1]  |Price:    0.99  |Vol:   10000
[ 2]  |Price:    1.01  |Vol:    2000
[ 3]  |Price:    1.03  |Vol:    4000
[ 4]  |Price:   1.035  |Vol:    4500
[ 5]  |Price:   1.047  |Vol:  100000

----- end of the book -----
```

**// 订单簿(10 档)初始化**

```
1001 A L B 0.98 1000
1002 A L B 0.957 6000
1001 A L B 0.959 4000
1003 A L A 0.99 10000
1005 A L B 0.958 5000
1004 A L A 1.047 100000
1006 A L B 0.96 3000
1005 A L A 1.035 4500
1009 A L A 1.1 5000
1333 A L A 1.01 2000
1450 A L A 1.03 4000
1500 A L B 0.965 2000
1001 A L B 0.94 10000
1002 A L B 0.93 200
1004 A L B 0.97 55555
1005 A L B 0.985 11111
1007 A L B 0.957 123456
1008 A L A 1.2 5555
1009 A L A 1.3 6000
1111 A L A 1.5 10000
1234 A L A 1.6 10000
```

```
---- 用户1234 您好!
---- 您的挂单回执是: ----
---- OrderId: 20

**** 买簿 ****
[ 1]  |Price:   0.985  |Vol:   11111
[ 2]  |Price:    0.98  |Vol:    1000
[ 3]  |Price:    0.97  |Vol:   55555
[ 4]  |Price:   0.965  |Vol:    2000
[ 5]  |Price:    0.96  |Vol:    3000
[ 6]  |Price:   0.959  |Vol:    4000
[ 7]  |Price:   0.958  |Vol:    5000
[ 8]  |Price:   0.957  |Vol:  129456
[ 9]  |Price:    0.94  |Vol:   10000
[10]  |Price:    0.93  |Vol:     200

**** 卖簿 ****
[ 1]  |Price:    0.99  |Vol:   10000
[ 2]  |Price:    1.01  |Vol:    2000
[ 3]  |Price:    1.03  |Vol:    4000
[ 4]  |Price:   1.035  |Vol:    4500
[ 5]  |Price:   1.047  |Vol:  100000
[ 6]  |Price:     1.1  |Vol:    5000
[ 7]  |Price:     1.2  |Vol:    5555
[ 8]  |Price:     1.3  |Vol:    6000
[ 9]  |Price:     1.5  |Vol:   10000
[10]  |Price:     1.6  |Vol:   10000

----- end of the book -----
```

2、撤单操作

```
1004 R 5
```

```
==== 撤单成功 ====

**** 买簿 ****
[ 1]  |Price:    0.98  |Vol:    1000
[ 2]  |Price:   0.965  |Vol:    2000
[ 3]  |Price:    0.96  |Vol:    3000
[ 4]  |Price:   0.959  |Vol:    4000
[ 5]  |Price:   0.958  |Vol:    5000

**** 卖簿 ****
[ 1]  |Price:    0.99  |Vol:   10000
[ 2]  |Price:    1.01  |Vol:    2000
[ 3]  |Price:    1.03  |Vol:    4000
[ 4]  |Price:   1.035  |Vol:    4500
[ 5]  |Price:     1.1  |Vol:    5000

----- end of the book -----
```

3、

①：下限价买单，与卖簿最优价进行撮合，买单无余量，卖簿最优价有余量

②：下限价买单，与卖簿最优价进行撮合，买单无余量，卖簿最优价也刚好无余量

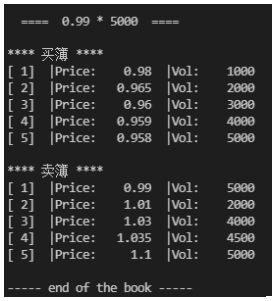
③：下限价卖单，与买簿最优价进行撮合，卖单无余量，买簿最优价有余量

④：下限价卖单，与买簿最优价进行撮合，卖单无余量，买簿最优价也刚好无余量

⑤：下限价买单,与卖簿最优价进行撮合,买单无余量,卖簿最优价全部交易，卖簿新的最优价被部分交易

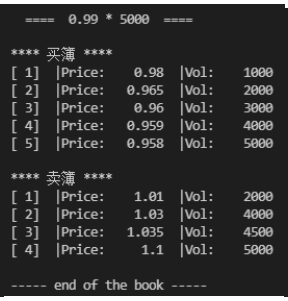
⑥：下限价卖单,与买簿最优价进行撮合，卖单无余量,买簿最优价全部交易,买簿新的最优价被部分交易

```
1111 A L B 0.995 5000  1112 A L B 0.994 5000
1005 A L A 0.979 500  1005 A L A 0.978 500
```



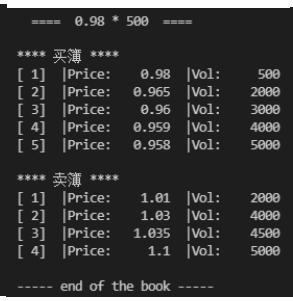
```
  ====  0.99 * 5000  ====

**** 买簿 ****
[ 1]  |Price:    0.98  |Vol:    1000
[ 2]  |Price:   0.965  |Vol:    2000
[ 3]  |Price:    0.96  |Vol:    3000
[ 4]  |Price:   0.959  |Vol:    4000
[ 5]  |Price:   0.958  |Vol:    5000

**** 卖簿 ****
[ 1]  |Price:    0.99  |Vol:    5000
[ 2]  |Price:    1.01  |Vol:    2000
[ 3]  |Price:    1.03  |Vol:    4000
[ 4]  |Price:   1.035  |Vol:    4500
[ 5]  |Price:     1.1  |Vol:    5000

----- end of the book -----
```

①



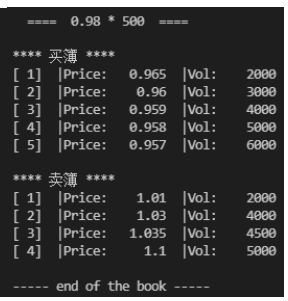
```
  ====  0.99 * 5000  ====

**** 买簿 ****
[ 1]  |Price:    0.98  |Vol:    1000
[ 2]  |Price:   0.965  |Vol:    2000
[ 3]  |Price:    0.96  |Vol:    3000
[ 4]  |Price:   0.959  |Vol:    4000
[ 5]  |Price:   0.958  |Vol:    5000

**** 卖簿 ****
[ 1]  |Price:    1.01  |Vol:    2000
[ 2]  |Price:    1.03  |Vol:    4000
[ 3]  |Price:   1.035  |Vol:    4500
[ 4]  |Price:     1.1  |Vol:    5000

----- end of the book -----
```

②



```
  ====  0.98 * 500  ====

**** 买簿 ****
[ 1]  |Price:    0.98  |Vol:     500
[ 2]  |Price:   0.965  |Vol:    2000
[ 3]  |Price:    0.96  |Vol:    3000
[ 4]  |Price:   0.959  |Vol:    4000
[ 5]  |Price:   0.958  |Vol:    5000

**** 卖簿 ****
[ 1]  |Price:    1.01  |Vol:    2000
[ 2]  |Price:    1.03  |Vol:    4000
[ 3]  |Price:   1.035  |Vol:    4500
[ 4]  |Price:     1.1  |Vol:    5000

----- end of the book -----
```

③

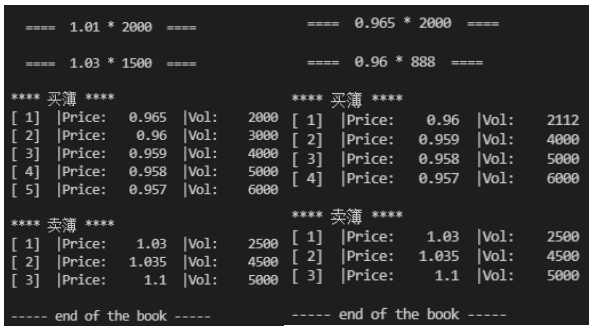


```
  ====  0.98 * 500  ====

**** 买簿 ****
[ 1]  |Price:   0.965  |Vol:    2000
[ 2]  |Price:    0.96  |Vol:    3000
[ 3]  |Price:   0.959  |Vol:    4000
[ 4]  |Price:   0.958  |Vol:    5000
[ 5]  |Price:   0.957  |Vol:    6000

**** 卖簿 ****
[ 1]  |Price:    1.01  |Vol:    2000
[ 2]  |Price:    1.03  |Vol:    4000
[ 3]  |Price:   1.035  |Vol:    4500
[ 4]  |Price:     1.1  |Vol:    5000

----- end of the book -----
```

④

2001 A L B 1.0344 3500

2002 A L A 0.96 2888



```
  ====  1.01 * 2000  ====

  ====  1.03 * 1500  ====

**** 买簿 ****
[ 1]  |Price:   0.965  |Vol:    2000
[ 2]  |Price:    0.96  |Vol:    3000
[ 3]  |Price:   0.959  |Vol:    4000
[ 4]  |Price:   0.958  |Vol:    5000
[ 5]  |Price:   0.957  |Vol:    6000

**** 卖簿 ****
[ 1]  |Price:    1.03  |Vol:    2500
[ 2]  |Price:   1.035  |Vol:    4500
[ 3]  |Price:     1.1  |Vol:    5000

----- end of the book -----
```

⑤



```
  ====  0.965 * 2000  ====

  ====  0.96 * 888  ====

**** 买簿 ****
[ 1]  |Price:    0.96  |Vol:    2112
[ 2]  |Price:   0.959  |Vol:    4000
[ 3]  |Price:   0.958  |Vol:    5000
[ 4]  |Price:   0.957  |Vol:    6000

**** 卖簿 ****
[ 1]  |Price:    1.03  |Vol:    2500
[ 2]  |Price:   1.035  |Vol:    4500
[ 3]  |Price:     1.1  |Vol:    5000

----- end of the book -----
```

⑥

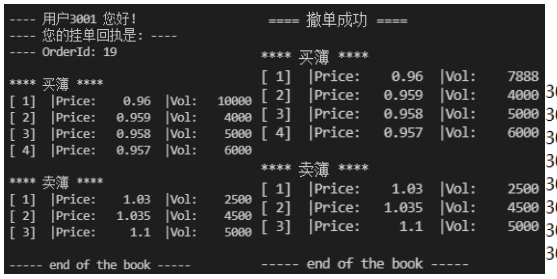
4、①：添加价格相同的订单

②：撤单 （撤掉同一价格下的某一单）

③：继续添加新订单(包含同一价格多个订单)（用于测试市价单）

3001 A L B 0.96 7888

1006 R 6



```
---- 用户3001 您好!
---- 您的挂单回执是: ----
---- OrderId: 19

**** 买簿 ****
[ 1]  |Price:    0.96  |Vol:   10000
[ 2]  |Price:   0.959  |Vol:    4000
[ 3]  |Price:   0.958  |Vol:    5000
[ 4]  |Price:   0.957  |Vol:    6000

**** 卖簿 ****
[ 1]  |Price:    1.03  |Vol:    2500
[ 2]  |Price:   1.035  |Vol:    4500
[ 3]  |Price:     1.1  |Vol:    5000

----- end of the book -----
```

①

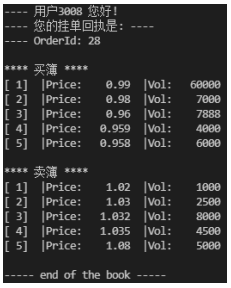


```
  ====  撤单成功  ====

**** 买簿 ****
[ 1]  |Price:    0.96  |Vol:    7888
[ 2]  |Price:   0.959  |Vol:    4000
[ 3]  |Price:   0.958  |Vol:    5000
[ 4]  |Price:   0.957  |Vol:    6000

**** 卖簿 ****
[ 1]  |Price:    1.03  |Vol:    2500
[ 2]  |Price:   1.035  |Vol:    4500
[ 3]  |Price:     1.1  |Vol:    5000

----- end of the book -----
```

②

3002 A L B 0.99 60000
3003 A L B 0.98 7000
3004 A L A 1.02 1000
3005 A L A 1.032 7000
3006 A L A 1.032 1000
3003 A L B 0.958 1000
3007 A L A 1.08 5000
3008 A L A 1.1 5000

③



```
---- 用户3008 您好!
---- 您的挂单回执是: ----
---- OrderId: 28

**** 买簿 ****
[ 1]  |Price:    0.99  |Vol:   60000
[ 2]  |Price:    0.98  |Vol:    7000
[ 3]  |Price:    0.96  |Vol:    7888
[ 4]  |Price:   0.959  |Vol:    4000
[ 5]  |Price:   0.958  |Vol:    6000

**** 卖簿 ****
[ 1]  |Price:    1.02  |Vol:    1000
[ 2]  |Price:    1.03  |Vol:    2500
[ 3]  |Price:   1.032  |Vol:    8000
[ 4]  |Price:   1.035  |Vol:    4500
[ 5]  |Price:    1.08  |Vol:    5000

----- end of the book -----
```

③

5、市价买单（无订单编号回执返回）

①市价买单示例 1：（卖簿最优价有余量）

②市价买单示例 2：（卖簿最优价刚好无余量）

③市价买单示例 3：（卖簿最优价无余量，产生新的卖簿最优价，且新的卖簿最优价部分交易）

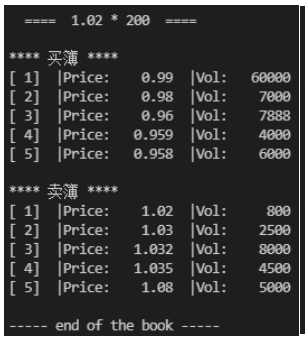
④继续添加卖单：

⑤市价买单示例 4：（单子太大，前 5 档全部成交，剩余撤销）

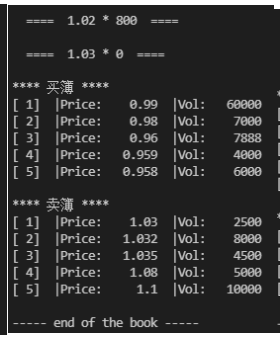
4001 A M B 200

4002 A M B 800

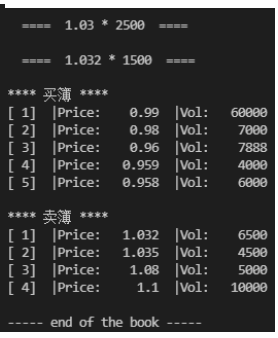
4003 A M B 4000



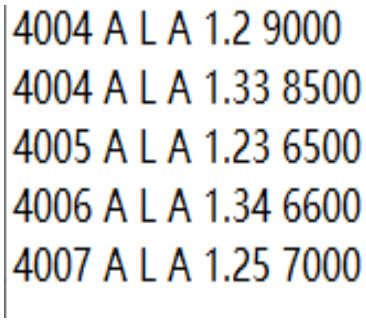

①



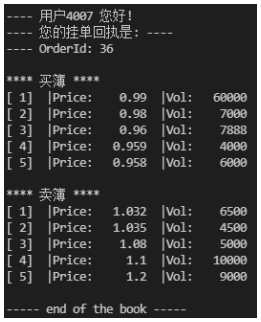

②




③

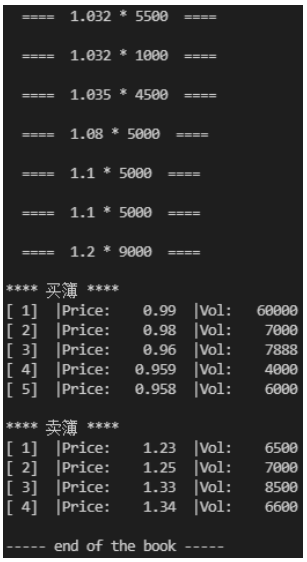



④




④

4008 A M B 50000




⑤

**6、 市价卖单（无订单编号回执返回）**

**①市价卖单示例 1：（买簿最优价有余量）**

**②市价卖单示例 2：（买簿最优价刚好无余量）**

**③市价卖单示例 3：（买簿最优价无余量，产生新的买簿最优价，且新的买簿最优价部分交易）**

④继续添加买单：

⑤市价卖单示例 4：（单子太大，前 5 档全部成交，剩余撤销）

```
5001 A M A 10000
```

```
==== 0.99 * 10000 ====

**** 买簿 ****
[ 1] |Price:   0.99  |Vol:   50000
[ 2] |Price:   0.98  |Vol:    7000
[ 3] |Price:   0.96  |Vol:    7888
[ 4] |Price:  0.959  |Vol:    4000
[ 5] |Price:  0.958  |Vol:    6000

**** 卖簿 ****
[ 1] |Price:   1.23  |Vol:    6500
[ 2] |Price:   1.25  |Vol:    7000
[ 3] |Price:   1.33  |Vol:    8500
[ 4] |Price:   1.34  |Vol:    6600

----- end of the book -----
```

①

```
5002 A M A 50000
```

```
==== 0.99 * 50000 ====

==== 0.98 * 0 ====

**** 买簿 ****
[ 1] |Price:   0.98  |Vol:    7000
[ 2] |Price:   0.96  |Vol:    7888
[ 3] |Price:  0.959  |Vol:    4000
[ 4] |Price:  0.958  |Vol:    6000
[ 5] |Price:  0.957  |Vol:    6000

**** 卖簿 ****
[ 1] |Price:   1.23  |Vol:    6500
[ 2] |Price:   1.25  |Vol:    7000
[ 3] |Price:   1.33  |Vol:    8500
[ 4] |Price:   1.34  |Vol:    6600

----- end of the book -----
```

②

```
5003 A M A 7888
```

```
==== 0.98 * 7000 ====

==== 0.96 * 888 ====

**** 买簿 ****
[ 1] |Price:   0.96  |Vol:    7000
[ 2] |Price:  0.959  |Vol:    4000
[ 3] |Price:  0.958  |Vol:    6000
[ 4] |Price:  0.957  |Vol:    6000

**** 卖簿 ****
[ 1] |Price:   1.23  |Vol:    6500
[ 2] |Price:   1.25  |Vol:    7000
[ 3] |Price:   1.33  |Vol:    8500
[ 4] |Price:   1.34  |Vol:    6600

----- end of the book -----
```

③

```
5004 A L B 0.97 1000
5005 A L B 0.95 40000
5006 A L B 0.94 10000
5007 A L B 0.93 2000
```

```
5009 A M A 50000
```

```
---- 用户5007 您好!
---- 您的挂单回执是: ----
---- OrderId: 44

**** 买簿 ****
[ 1] |Price:   0.97  |Vol:    1000
[ 2] |Price:   0.96  |Vol:    7000
[ 3] |Price:  0.959  |Vol:    4000
[ 4] |Price:  0.958  |Vol:    6000
[ 5] |Price:  0.957  |Vol:    6000

**** 卖簿 ****
[ 1] |Price:   1.23  |Vol:    6500
[ 2] |Price:   1.25  |Vol:    7000
[ 3] |Price:   1.33  |Vol:    8500
[ 4] |Price:   1.34  |Vol:    6600

----- end of the book -----
```

④

```
==== 0.97 * 1000 ====

==== 0.96 * 7000 ====

==== 0.959 * 4000 ====

==== 0.958 * 5000 ====

==== 0.958 * 1000 ====

==== 0.957 * 6000 ====

**** 买簿 ****
[ 1] |Price:   0.95  |Vol:   40000
[ 2] |Price:   0.94  |Vol:   10000
[ 3] |Price:   0.93  |Vol:    2000

**** 卖簿 ****
[ 1] |Price:   1.23  |Vol:    6500
[ 2] |Price:   1.25  |Vol:    7000
[ 3] |Price:   1.33  |Vol:    8500
[ 4] |Price:   1.34  |Vol:    6600

----- end of the book -----
```

⑤

## 四、【需要添加/改正的内容】

①：对于每一次交易，应该给涉及成交的用户都返回一个回执信息，告诉他们，“订单被交易了多少数量”、“订单已经被完全交易”。

②：撤单操作：应该添加一些判断，例如，此单号是否存在，此单号是否还在委托序列中。

③：交易所不是每一个单子都接受的。比如，Price 应该统一规定小数点位数，不然无法规避“99.99999999”等恶意操作。对于 Volumn，规定最小下单数量、规定整数下单等规则。

④：对于信息展示，当前价格、即最后一笔成交的价格未显示。
⑤：关于用户维护方面，我还未来得及拓展。其实输入文件中的 User_Id 对整个程序是没有影响的。后续初步想法：构建一个用户类，存储与更新用户信息，如持仓量、挂单编号、回执信息等。

**五、【项目收获与感谢】**

①：（1）写第一个版本的时候，使用的数据结构是数组，构造起订单簿非常困难。源于第一，无法知道添加订单的总数量；第二，要满足时间优先、价格优先的规则，这个数组就要不停地维护；第三，代码重用少、变量间的关系弱，代码写起来比较困难。（2）后来使用 map 去写，很多问题迎刃而解。因为 map 内部自建一棵红黑树，具有对数据自动排序的功能，map 中所有数据都是有序的。于是，我构建了一个<double,map<long long,Order>>类型，即“价格”——>“时间”——>“订单”的映射。在遍历过程中，即自动按照“价格低——>价格高”“同一价格下，时间先——>时间后”去锁定订单，对于交易撮合的部分，非常方便。
②：有两个 Bug 找了很久。第一个系在遍历 map 过程中，同时删除了 map 中的元素，导致迭代错误。第二个系重写构造函数后，没有将默认构造函数写上，这个不知道为什么几度使服务器崩溃(junfu 说是 CPU 过载)，这里 mark 一下，之后回到学校我再深入研究一下。
③：在此次训练营中，我收获颇丰，同时也深刻认识到自己 C++知识的贫瘠。由此 OrderBook 项目，我初探了股票交易系统是怎样的、大致了解了用户、交易所、证券公司之间的关系、“他们是怎样一环一环相互运作的”。非常感谢肖总、jiaman 姐等同事朋友针对我的项目，提出了很多宝贵的意见。如：如果此项目落地、要交付用户使用，那么只能对用户提供一个信息输入的接口，根据用户提供的信息再回馈。我代码中 OrderBook_Engine 类的 Order_Update 方法是公开的，这样用户就可以进行攥写，不安全、业务也不会这样写。封装的程度要更上一层。