Stock Exchange Simulator

==================

An exchange is a place where transactions (Trade) are made between buyers and sellers. Buyers want to buy something, and sellers want to sell something. A purchase/sale transaction is made if the seller and the buyer are satisfied with the terms of the transaction - for example, the price and quantity are acceptable to both parties.

Sellers try to sell their goods at the highest possible price, and buyers try to buy at the lowest possible price. On the exchange, the desire to buy or sell is expressed in the submission of electronic applications (order), each of which indicates:

a) unique application number(ID)

b) buy or sell(side) - an indication to buy or sell ((B)uy or (S)ell)

c) the name of the security (Instrument)

d) quantity(Qty) - how much to buy/sell

e) purchase/sale price - the desire to buy/ sell at a price not worse than that indicated in the application. For purchase orders, the price must be greater than or equal to the specified one, for a sale order, the price must be equal to or less than the specified one.

**For example:**

O,4,B,Apples,10,256.12 -> application ((O)rder) with number 4 in which someone wants to buy((B)uy) Apples in the amount of 10 kg at the price of 256.12 RUB/kg or O,50,S,Apples,4,255.5 -> application ((O)rder) with the number 50 in which someone wants to sell((S)ell) Apples in the amount of 4 kg at a price of 255.5 rubles / kg.

If the purchase/sale condition satisfies both parties, then a Trade is made. In this example, the market conditions satisfy the requirements for making a deal:

a) there is a sufficient amount in this case min(Buy Qty, Sell Qty) = min(10,4) = 4

b) the buyer's price allows the seller to fulfill the condition to sell no worse than his selling price (Sell Price > = Buy Price), and the buyer has the opportunity to buy at the price he stated, which also satisfies the purchase condition – to buy no worse

Thus, the transaction price will be equal to 256.12 and the exchange will generate the final transaction:

T,1,B,4,50,4,256.12 -> a transaction ((T)grade) has been made for the purchase with the number 1 for the purchase between orders 4 and 50 in the amount of 4 kg at the price of 256.12

It should be noted that it is the buyer's price that is chosen as the transaction price, because he submitted the order earlier than the seller, which follows from the fact that the buyer's application number is less than the seller's order number.

After the transaction is completed, the seller's request is deleted from the exchange, because it was executed in full, and the buyer's request remains on the exchange, but with a balance equal to 10 - 4 = 6.

If there are no conditions for making transactions, then applications remain on the exchange until they are withdrawn. In order to withdraw an application, it is necessary to send an order to the exchange to cancel the application:

C,4 -> cancel((C)ancel) application number 4

In real conditions, there are many sellers and buyers on the exchange, so the algorithm for executing the application will be as follows:

**For a new purchase request:**

1) All sales orders are sorted by price in the ascending direction and transactions are made primarily with orders with the lowest price in order to satisfy the best buy rule (at the lowest price). If there are several applications for sale at the same price, then the applications with the lowest application number (ID) are executed first - the FIFO rule (First In First Out)

2) If there are no orders for transactions, then the purchase order remains on the exchange

**For a new sale request:**

1) All purchase orders are sorted by price in descending order and transactions are made primarily with orders to buy at the highest price in order to satisfy the best sale rule (at the highest price). If there are several purchase orders at the same price, then first of all the orders with the lowest order number (ID) are executed - the FIFO rule (First In First Out)

2) If there are no locks for making transactions, then the sale request remains on the exchange

**Example:**

1. The following purchase orders are active on the exchange:

Oh,1,B,Apples,10,250.12

Oh,2,B,Apples,8,249.10

Oh,3,B,Apples,5,250.12

Oh,4,B,Apples,15,250.6

2. A new application is submitted to the exchange for the sale of 31 kg of apples:

O,5,S,Apples,31,248.5

3. We sort the purchase orders in descending order by price + number of the order:

O,4,B,Apples,15,250.6

Oh,1,B,Apples,10,250.12

Oh,3,B,Apples,5,250.12

Oh,2,B,Apples,8,249.10

4. we see that a number of orders satisfy the conditions for making transactions, so the exchange algorithm generates transactions:

T,1,B,4,5,15,250.6

T,2,B,1,5,10,250.12

T,3,B,3,5,5,250.12

T,4,B,2,5,1,249.10

5. The purchase order remains on the exchange but with a balance of 7:

O,2,B,Apples,7,249.10

6. All other applications are deleted because they have been fully executed

**Task**

=======

To develop an algorithm for an exchange simulator:

a) the exchange trades only apples

b) the simulator receives input signals and executes them:

1) O,<OID>,<Side>,<Qty>,<Price> - request to buy ((B)uy) or sell

((S)ell):

• <OID> is the unique identifier of the application (number). In fact, the OID

is generated by the exchange itself, so in our case it is always an increasing number

• <Side> - purchase(B)/sale(S)

• <Qty> - buy/sell quantity. An integer

• <Price> - the purchase/sale price. Floating point number – maximum 2

the decimal point

2) C,<OID> - cancellation of a previously placed order by number <OID>. If an attempt is made to cancel an application that has already been canceled or executed, the cancellation operation is ignored.

c)the simulator at the output should generate transactions and print:

1) T,<ID>,<Side>,<OID1>,<OID2>,<Trade Qty>,<Trade Price> - transaction:

• <ID> - unique identifier of the transaction (a monotonously increasing number)

• <Side> - the side of the transaction. Buy or Sell

• <OID1> - the identifier of the order with which the transaction was made (

previously issued)

• <OID2> is the identifier of the order that initiated the transaction. <OID1>

is always less than <OID2>

• <Trade Qty> - quantity of goods purchased/sold

• <Trade Price> - the price of the transaction

2) X,<OID> - information that the application has been canceled

d) the algorithm should be a console application that accepts a stream (file or stdin) of requests (O) and a cancellation operation(C) and prints (stdout or file) transactions(T) and cancellation confirmations(X)

e) it is suggested to use a file for testing input.txt with applications and cancellation operations and a file output.txt with transactions and confirmations of cancellation of applications