The Engineering team at Deep-Rooted.Co consists of people who are passionate about code. Please find the coding challenge which helps us assess your coding skill which helps us position you in the company for success and fix your compensation.

Following are the details of the coding round

- Attachment has the problem statement demand-supply-problem.txt.
- Solution can be provided in Kotlin or Java or Clojure.
- We are interested in the design, coding style of the solution
- The program must have good tests that verify the correctness.
- Standard Java libraries can be used. Please use Java 11 or above.
- Please do not use third-party libraries.
- JUnit or equivalent Kotlin test framework must be used for testing.
- If necessary, include a brief explanation of your design and assumptions along with your code.

Please zip and send the code to us before**.**

Please do not share the problem and solution with others or publicly.
Please honor confidentiality of the problem and solution.

Regards
Srikanth Seshadri

Chief Technology Officer
B.Sc Computer Science BMS college of Engg
Ex-Navi, Sahaj, Thoughtworks
20+ years of building product and engineering teams

Code Challenge!!

Demand-Supply Matching

Implement a demand-supply matching program for an online market maker.

1. Farmers/Traders publish the availability of the produce with details - quantity and price.These are supply orders.
2. Customers/Traders publish their requirement for the produce - quantity and the best price they can offer. These are demand orders.

All the requirements and the availability are stored in the demand-supply ledger.

The application will match the demand with the supply in the ledger. Whenever a new supply or demand is published; matching is done.

If the requirement cannot be satisfied it continues to remain in the ledger. No expiry is supported.

The program must follow the rules below

1. Priority must be given to "lower supply price - higher demand price" matching. Hence maximizing the profit for the market maker.
2. The supplier is always given the price he has asked for regardless of the price offered on the demand side.
3. Within the same supply/demand price, first-in-first out rule on time must be followed. First supply must be matched to the first demand.
4. A trade is generated when a buy price is greater than or equal to a sell price. As mentioned earlier the trade is recorded at the price of the supply regardless of the price of the demand.

Write a program that accepts supply/demand orders from standard input and writes trades to standard output. Do not prompt for input. Please write tests the demonstrate the correct ness of the functionality

Example 1:
The following input format:
format: <order-id> <time> <produce> <price/kg> <quantity in kg>:

s1 09:45 tomato 24/kg 100kg
s2 09:46 tomato 20/kg 90kg
d1 09:47 tomato 22/kg 110kg
d2 09:48 tomato 21/kg 10kg
d3 09:49 tomato 21/kg 40kg
s3 09:50 tomato 19/kg 50kg

order-id starting with
- `s` are supply orders
- `d` are demand orders

Should produce the following output
format: <demand-order-id> <supply-order-id> <price/kg> <quantity in kg>):

d1 s2 20/kg 90kg
d1 s3 19/kg 20kg
d2 s3 19/kg 10kg
d3 s3 19/kg 20kg

Example 2:

Input:
d1 09:47 tomato 110/kg 1kg
d2 09:45 potato 110/kg 10kg
d3 09:48 tomato 110/kg 10kg
s1 09:45 potato 110/kg 1kg
s2 09:45 potato 110/kg 7kg
s3 09:45 potato 110/kg 2kg
s4 09:45 tomato 110/kg 11kg

Output:
d2 s1 110/kg 1kg
d2 s2 110/kg 7kg
d2 s3 110/kg 2kg
d1 s4 110/kg 1kg
d3 s4 110/kg 10kg