* Your program exposes two functions: addOrder() and newTick(). An external program calls these functions to send in new orders and ticks.
  + Each incoming order is stored.
  + Each incoming tick is used to process stored orders. When the relevant stored orders have been processed, that tick is discarded.
  + Order structure: { symbol: 'GOOG', type: 'BL', order\_price: 123.45, quantity: 5 }
  + Tick structure: { symbol: 'GOOG', price: 120.99, timestamp: ... }
* Orders are stored until they are filled by a tick. When a tick comes in, all stored orders matching any of the following conditions are processed.
  + When a tick price is below a Buy Limit order's price, the Buy Limit order is considered filled and is removed from order storage.
  + When a tick price is above a Buy Stop order's price, the Buy Stop order is considered filled and is removed from order storage.
  + etc.
* Orders come in occasionally, while ticks come in regularly and very quickly.
* How would you store orders and process incoming ticks for maximum processing speed? Focus on the data structure for storing orders and the algorithm for processing ticks that come in.
* Write the code for addOrder(), newTick(), and any other variables/structures/functions you deem suitable.

**Example**

**Example:**

**Input:**

* Pending Orders:
  + Order 1: AAPL, Buy Limit, Price: 100, Quantity: 10
  + Order 2: AAPL, Buy Limit, Price: 105, Quantity: 5
  + Order 3: AAPL, Buy Limit, Price: 102, Quantity: 15
  + Order 4: AAPL, Buy Limit, Price: 98, Quantity: 15
  + Order 5: GOOG, Buy Limit, Price: 200, Quantity: 10
  + Order 6: GOOG, Buy Limit, Price: 201, Quantity: 15
  + Order 7: GOOG, Buy Limit, Price: 198, Quantity: 10
* Ticks:
  + [AAPL, Time: 08:58, Price: 106]
  + [AAPL, Time: 09:00, Price: 104]
  + [AAPL, Time: 09:01, Price: 102]
  + [AAPL, Time: 09:02, Price: 98]
  + [GOOG, Time: 09:03, Price: 200]

**Output:**

Order 2 executed at 09:00, Price: 104, Quantity: 5

Order 3 executed at 09:01, Price: 102, Quantity: 15

Order 1 executed at 09:02, Price: 98, Quantity: 10

Order 4 executed at 09:02, Price: 98, Quantity: 15

Order 6 executed at 09:03, Price: 200, Quantity: 15

Order 5 executed at 09:03, Price: 200, Quantity: 10

**Explanation:**

* At 09:00, the price is 104, triggering the execution of the Buy Limit order (Order 2).
* At 09:01, the price is 102, triggering the execution of the Buy Limit order (Order 3).
* At 09:03, the price is 98, triggering the execution of the Buy Limit order (Order 1) and Buy Limit order (Order 4).
* At 09:03, the price is 200, triggering the execution of the Buy limit order (Order 5 and 6).