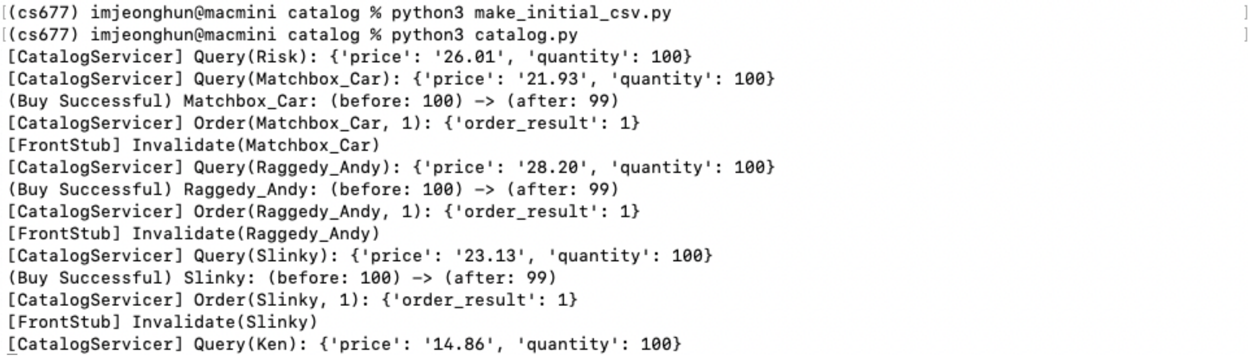
Output File

*All messages that are printed using '[' and ']' follows the following format:
['Class name of stub instance'] 'request name'('parameters'): 'request result'

- If a class name ends with 'Stub', then the instance is the client side of the RPC call.
- If a class name ends with 'Servicer', then the instance is the server side of the RPC call.

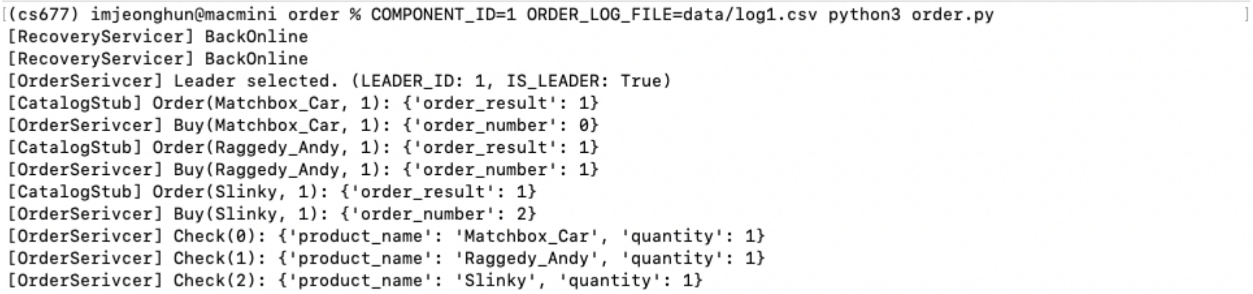
*Ping RPC Calls are not printed out because they are too frequent. (Repeated every one second between the front-end component and the leader order component.)

- catalog component output



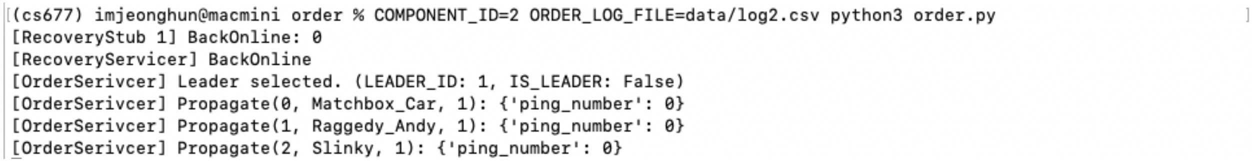
```
(cs677) imjeonghun@macmini catalog % python3 make_initial_csv.py
(cs677) imjeonghun@macmini catalog % python3 catalog.py
[CatalogServicer] Query(Risk): {'price': '26.01', 'quantity': 100}
[CatalogServicer] Query(Matchbox_Car): {'price': '21.93', 'quantity': 100}
(Buy Successful) Matchbox_Car: (before: 100) -> (after: 99)
[CatalogServicer] Order(Matchbox_Car, 1): {'order_result': 1}
[FrontStub] Invalidate(Matchbox_Car)
[CatalogServicer] Query(Raggedy_Andy): {'price': '28.20', 'quantity': 100}
(Buy Successful) Raggedy_Andy: (before: 100) -> (after: 99)
[CatalogServicer] Order(Raggedy_Andy, 1): {'order_result': 1}
[FrontStub] Invalidate(Raggedy_Andy)
[CatalogServicer] Query(Slinky): {'price': '23.13', 'quantity': 100}
(Buy Successful) Slinky: (before: 100) -> (after: 99)
[CatalogServicer] Order(Slinky, 1): {'order_result': 1}
[FrontStub] Invalidate(Slinky)
[CatalogServicer] Query(Ken): {'price': '14.86', 'quantity': 100}
```

-order component 1 output



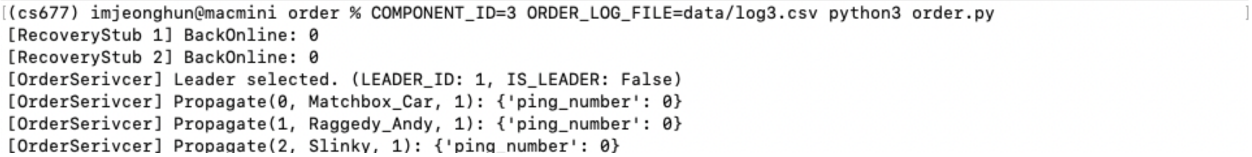
```
(cs677) imjeonghun@macmini order % COMPONENT_ID=1 ORDER_LOG_FILE=data/log1.csv python3 order.py
[RecoveryServicer] BackOnline
[RecoveryServicer] BackOnline
[OrderSerivcer] Leader selected. (LEADER_ID: 1, IS_LEADER: True)
[CatalogStub] Order(Matchbox_Car, 1): {'order_result': 1}
[OrderSerivcer] Buy(Matchbox_Car, 1): {'order_number': 0}
[CatalogStub] Order(Raggedy_Andy, 1): {'order_result': 1}
[OrderSerivcer] Buy(Raggedy_Andy, 1): {'order_number': 1}
[CatalogStub] Order(Slinky, 1): {'order_result': 1}
[OrderSerivcer] Buy(Slinky, 1): {'order_number': 2}
[OrderSerivcer] Check(0): {'product_name': 'Matchbox_Car', 'quantity': 1}
[OrderSerivcer] Check(1): {'product_name': 'Raggedy_Andy', 'quantity': 1}
[OrderSerivcer] Check(2): {'product_name': 'Slinky', 'quantity': 1}
```

-order component 2 output



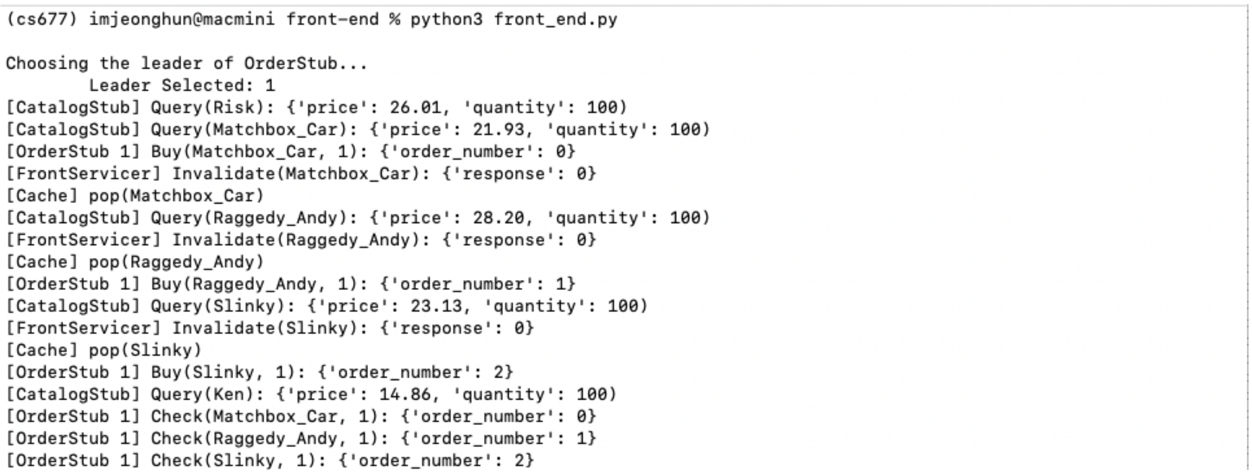
```
(cs677) imjeonghun@macmini order % COMPONENT_ID=2 ORDER_LOG_FILE=data/log2.csv python3 order.py
[RecoveryStub 1] BackOnline: 0
[RecoveryServicer] BackOnline
[OrderSerivcer] Leader selected. (LEADER_ID: 1, IS_LEADER: False)
[OrderSerivcer] Propagate(0, Matchbox_Car, 1): {'ping_number': 0}
[OrderSerivcer] Propagate(1, Raggedy_Andy, 1): {'ping_number': 0}
[OrderSerivcer] Propagate(2, Slinky, 1): {'ping_number': 0}
```

-order component 3 output



```
(cs677) imjeonghun@macmini order % COMPONENT_ID=3 ORDER_LOG_FILE=data/log3.csv python3 order.py
[RecoveryStub 1] BackOnline: 0
[RecoveryStub 2] BackOnline: 0
[OrderSerivcer] Leader selected. (LEADER_ID: 1, IS_LEADER: False)
[OrderSerivcer] Propagate(0, Matchbox_Car, 1): {'ping_number': 0}
[OrderSerivcer] Propagate(1, Raggedy_Andy, 1): {'ping_number': 0}
[OrderSerivcer] Propagate(2, Slinky, 1): {'ping_number': 0}
```

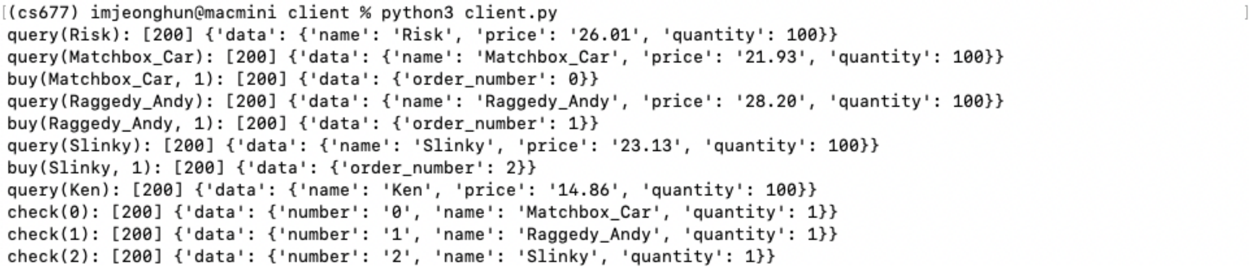
-front-end output



```
(cs677) imjeonghun@macmini front-end % python3 front_end.py

Choosing the leader of OrderStub...
        Leader Selected: 1
[CatalogStub] Query(Risk): {'price': 26.01, 'quantity': 100)
[CatalogStub] Query(Matchbox_Car): {'price': 21.93, 'quantity': 100)
[OrderStub 1] Buy(Matchbox_Car, 1): {'order_number': 0}
[FrontServicer] Invalidate(Matchbox_Car): {'response': 0}
[Cache] pop(Matchbox_Car)
[CatalogStub] Query(Raggedy_Andy): {'price': 28.20, 'quantity': 100)
[FrontServicer] Invalidate(Raggedy_Andy): {'response': 0}
[Cache] pop(Raggedy_Andy)
[OrderStub 1] Buy(Raggedy_Andy, 1): {'order_number': 1}
[CatalogStub] Query(Slinky): {'price': 23.13, 'quantity': 100)
[FrontServicer] Invalidate(Slinky): {'response': 0}
[Cache] pop(Slinky)
[OrderStub 1] Buy(Slinky, 1): {'order_number': 2}
[CatalogStub] Query(Ken): {'price': 14.86, 'quantity': 100)
[OrderStub 1] Check(Matchbox_Car, 1): {'order_number': 0}
[OrderStub 1] Check(Raggedy_Andy, 1): {'order_number': 1}
[OrderStub 1] Check(Slinky, 1): {'order_number': 2}
```

-client output



```
(cs677) imjeonghun@macmini client % python3 client.py
query(Risk): [200] {'data': {'name': 'Risk', 'price': '26.01', 'quantity': 100}}
query(Matchbox_Car): [200] {'data': {'name': 'Matchbox_Car', 'price': '21.93', 'quantity': 100}}
buy(Matchbox_Car, 1): [200] {'data': {'order_number': 0}}
query(Raggedy_Andy): [200] {'data': {'name': 'Raggedy_Andy', 'price': '28.20', 'quantity': 100}}
buy(Raggedy_Andy, 1): [200] {'data': {'order_number': 1}}
query(Slinky): [200] {'data': {'name': 'Slinky', 'price': '23.13', 'quantity': 100}}
buy(Slinky, 1): [200] {'data': {'order_number': 2}}
query(Ken): [200] {'data': {'name': 'Ken', 'price': '14.86', 'quantity': 100}}
check(0): [200] {'data': {'number': '0', 'name': 'Matchbox_Car', 'quantity': 1}}
check(1): [200] {'data': {'number': '1', 'name': 'Raggedy_Andy', 'quantity': 1}}
check(2): [200] {'data': {'number': '2', 'name': 'Slinky', 'quantity': 1}}
```