

We love hearing from happy LimeBike users about the usefulness of our bike baskets, which are great for carrying important items during bike rides. In order to better understand our service's impact on the movement of goods, please use your favorite programming language to write a class `ItemCounter` that implements the two following functions:

```
# Processes a Ride object that contains metadata about the ride, including:
# start time, end time, and bike basket items.
process_ride(ride)

# Prints a summary of the numbers of each type of item in transit during any
# given time interval. Excludes time intervals during which no items were in
# transit.
print_items_per_interval()
```

As an example, suppose we processed the following rides with an `ItemCounter` instance:

Ride 1:

Start/End times: 07:00-07:30
Items: 2 apples, 1 brownie

Ride 2:

Start/End times: 07:10-08:00
Items: 1 apple, 3 carrots

Ride 3:

Start/End times: 07:20-07:45
Items: 1 apple, 2 brownies, 4 diamonds

When we call `ItemCounter#print_items_per_interval()`, something like the following should be printed:

```
07:00-07:10 -> 2 apples, 1 brownie
07:10-07:20 -> 3 apples, 1 brownie, 3 carrots
07:20-07:30 -> 4 apples, 3 brownies, 3 carrots, 4 diamonds
07:30-07:45 -> 2 apples, 2 brownies, 3 carrots, 4 diamonds
07:45-08:00 -> 1 apple, 3 carrots
```

Note that the printed time intervals here are sorted and continuous. Don't worry too much about the input and output formats--just demonstrate how to use your functions in a few lines of code, using hard-coded `Time` objects if that's easiest for you. In addition to the `ItemCounter` class, free to define your own classes or structs to help with your implementation.

Your code will be evaluated on correctness, code cleanliness, and efficiency (time / space complexity). Thanks a lot for taking the time to do the coding exercise, and we appreciate your interest in LimeBike!