

← [course home \(/table-of-contents#section_greedy_question_cafe-order-checker\)](/table-of-contents#section_greedy_question_cafe-order-checker)

My cake shop is so popular, I'm adding some tables and hiring wait staff so folks can have a cute sit-down cake-eating experience.

I have two registers: one for take-out orders, and the other for the other folks eating inside the cafe. All the customer orders get combined into one list for the kitchen, where they should be handled first-come, first-served.

Recently, some customers have been complaining that people who placed orders after them are getting their food first. Yikes—that's not good for business!

To investigate their claims, one afternoon I sat behind the registers with my laptop and recorded:

- The take-out orders as they were entered into the system and given to the kitchen. (`take_out_orders`)
- The dine-in orders as they were entered into the system and given to the kitchen. (`dine_in_orders`)
- Each customer order (from either register) as it was finished by the kitchen. (`served_orders`)

Given all three lists, write a function to check that my service is first-come, first-served. All food should come out in the same order customers requested it.

We'll represent each customer order as a unique integer.

As an example,

Take Out Orders: [1, 3, 5]

Dine In Orders: [2, 4, 6]

Served Orders: [1, 2, 4, 6, 5, 3]

would not be first-come, first-served, since order 3 was requested before order 5 but order 5 was served first.

But,

Take Out Orders: [17, 8, 24]

Dine In Orders: [12, 19, 2]

Served Orders: [17, 8, 12, 19, 24, 2]

would be first-come, first-served.

Note: Order numbers are arbitrary. They do **not** have to be in increasing order.

Want more coding interview help?

Check out **interviewcake.com** for more advice, guides, and practice questions.