

Ellevation Education Collaboration Exercise

Duration: Ahead of time, up to 2 hours; in-person, 45 minutes

Summary: Design an elevator controller for a ten floor building with a single elevator. Write a simulator that can model the movement of the elevator through time in response to multiple user requests.

Expectations: Please arrive ready to spend the first 10-15 minutes of the session presenting your solution. Explain any important design decisions and any assumptions that you made. The reviewers will then ask questions, delve into areas of interest, and present new requirements that you will be asked to incorporate into your design.

We ask that you bring runnable code to the interview.

There is no technology stack requirement — please use the programming language and design approaches with which you are comfortable. You may be asked to modify your code or write additional code during the session.

We value simple, working code over a partially-implemented, complex solution.

You are welcome to provide copies of your code and any documentation or diagrams in advance, although lengthy slide decks are discouraged.

We believe that having you write code ahead of time allows us to have a better conversation during our time together. Our goals are to get a sense of your programming skills and understand how you solve problems and collaborate with a team of like-minded engineers.

Please let us know if you have any questions about the exercise.

Problem Description

Design a simulator for an elevator control system. You may assume (for now) that the building has ten floors and only one elevator car. In the elevator lobby on each floor there are two buttons, one to call the elevator in the *Up* direction and one for the *Down* direction. When the passenger enters the elevator car, they press a button to indicate their desired destination floor.

It should model user interactions from outside the elevator (pressing up/down buttons) and inside the elevator (pressing a button for a floor). To simplify the design, assume that the elevator moves up (or down) one floor per *time* unit. The process of opening and closing the doors, and passengers entering and exiting the elevator car, happens instantaneously (takes no time).

We will provide you test cases that involve multiple user requests spaced out in time and ask that you demonstrate how your elevator responds to those requests. Here is an example test case:

- The elevator car begins at the lobby (floor 1).
- At time 0, a user on floor 3 presses the “Down” request button. Their destination is floor 2.
- At time 1, a user on floor 10 presses the “Down” request button. Their destination is floor 1.

It is up to you to determine how to serve those requests, but we expect the simulator to output the current state of the elevator at each “tick” of time. It might look something like:

```
{ time: 1, direction: Up, floor: 2 }
```

We also expect the simulator to process passengers embarking/disembarking based on requested stops - simulating the passenger entering the elevator, pressing an internal floor button and ultimately departing at their destination floor.

We'll have you walk through your code and the underlying assumptions you made and then run a few additional test scenarios. Time permitting, we will discuss how you would alter your approach as new requirements or constraints are added, such as multiple cars, express elevators, service floors, etc.