

Design and program an elementary elevator control system. The focus is only on moving and tracking elevators in a building - real world concerns like weight limits, fire control, overrides, holds, etc., are beyond the scope of the program.

Parameters for the program:

- the building has 10 floors
- there are four elevators
- it takes 10 seconds for an elevator car to move from one floor to the next
- when a car stops on a floor, it takes 10 seconds for passengers to enter / leave and then the car is ready to move again

Write a program that generates random calls for the elevator on floors throughout the building. The elevator cars will move to pick up passengers and disembark them. The algorithm can be simple and naive, but in general, an "up" elevator should keep going up until it has no more passengers and the same for a "down" elevator - an elevator shouldn't yo-yo up and down between floors while still containing passengers. A demonstrated optimized algorithm is extra credit, but not necessary. The program should indicate the relative position of the elevator cars (car 1 is on floor 3, car 2 is on floor 10) and user actions ("down" request on floor 4 received, "up" request on floor 7 received); this can be as simple as console logging - more complicated UI is extra credit.

No one expects this to be "production ready". It's a simple programming exercise. Don't get complicated or worry about one-offs and special cases. However, do treat this as "production like" and not a one-time script; assume it is code that will be reviewed, must be maintained, will be augmented later, etc. Give us an idea of what your code will be when submitted on an actual real-world project. We're not looking for 'clever' code or a full application, but rather an example of your coding style. Clean code, appropriate comments and adequate test coverage are appreciated.

Feel free to use whatever programming language you prefer, but C# and Typescript are used extensively at our company.

Notes:

There's no time limit and candidates can choose exactly how to implement it (language, tools, etc.). Candidates should feel free to use the internet and other resources to assist them while writing it just as they would be writing actual code for work. However, the work needs to be their own – **no pair programming with a friend, etc.** Once we receive the code, we'll review it and schedule a follow-up session to discuss their specific design and programming choices with the candidates.