Elevator Simulator Requirements

An Early Draft

# Customer Requirements

Provide code that simulates an elevator.  You are free to use any language (recommend C++).

Upload to github for discussion during interview.

Document all assumptions and any features that weren’t implemented.

The result should be an executable, or script, that can be run with the following inputs and generate the following outputs.

* Inputs: <starting floor> [list of floors to visit]  (e.g elevator start=12 floor=2,9,1,32)
* Outputs: <total travel time> [floors visited in order] (e.g 560 12,2,9,1,32)

Program Constants: Single floor travel time: 10 seconds

# Derived Requirements

1. The software shall simulate an elevator
   1. The software shall set the starting floor
      1. The default starting floor shall be the first floor
      2. The software shall allow the user to set a custom starting floor
   2. The software shall simulate travel between floors
      1. The software shall simulate travel to each floor in a list
      2. The software shall keep track of total time spend traveling between floors
      3. The single floor travel time shall be defined as 10 seconds
2. The software shall contain a command line user interface.
   1. The UI should be able top run a default test run
   2. The UI shall be able to accept user input to run an elevator simulation
      1. The UI shall have a defined input as follows
         1. The elevator starting floor shall be a single integer
         2. The list of floors to visit shall be a string containing integers separated by commas
      2. The software shall have a defined output of a string containing the total travel time followed by a space then the list of floors visited where each floor is separated by comma
3. The software shall be placed under configuration management
   1. Github shall be used
   2. An executable or script shall be provided for release to customer

# Assumptions:

1. Floor names can change between buildings and they almost never have a floor zero. To simplify the code, our floor labeling schema will be:
   1. Floor Zero is the ground floor
   2. A positive floor value will indicate number of floors above ground level
   3. A negative floor number will indicate number of floors below ground level (basement floors)
2. The total operating type of the elevator will be less than 4,398,046,511,104 seconds ≈ 139,369 years. This is to support using a double to track total time.

# Notes:

1. The Given example of output (e.g 560 12,2,9,1,32) appears to be correct
   1. Floors traveled are 10,7,8,31 which when summed equals 56
   2. 56 x 10 (single floor travel time) = 560
2. An elevator will track its own total travel times (?for maintenance reasons?). Potentially rounding to the nearest second could create an inaccurate total travel time. This could potentially cause a maintenance related safety issue. This is a quick and dirty analysis to see if it would be better to use a double for tracking travel time.
   1. A double has 52 fraction bits if we want at least 1/100 of a second precision we can dedicate 7 bits for fractions of a second (2^7 = 128 values). This leaves 42 bits for integer values of seconds.
   2. This means we can track a total time of 2^42 = 4,398,046,511,104 seconds = 139,369 years while maintaining at least 1/100 second precision. That appears to be acceptable.