Emotive.io 48 Hour Take Home Exam

Please Answer Questions 1 and 2. Please answer and be ready to discuss your approach to question 3 during your on-site in detail.

1. Emotive uses Twilio to power the sending of messages on our back-end. We use a function called validate_bulk_messages(copilot_ids) that takes as an argument a list of integer values called Co-Pilot IDs (copilot_ids) that Twilio uses to send out messages. Recently Twilio is having a major bug and Emotive has been getting duplicate IDs for different messages that should be sent to different people. Co-Pilot IDs should be unique for the sending of messages in bulk. Write a function that takes in a list of Co-Pilot IDs and increments each one by one until the list you get is a list of unique Co-Pilot IDs and the sum of these values is the minimum such value possible for unique Co-Pilot IDs (after incrementing to get a unique list).

   For example the list we get form Twilio is copilot_ids = [122, 144, 122, 144, 181]. However your function should return [122, 123, 144, 145, 181].

   As another example we get from Twilio a list as follows [13458, 13890, 13890, 144568, 144568]. However your function should return [13458, 13890, 13891, 144568, 144569] as the unique list copilot_ids.

2. Jeff Bezos has done a fantastic job running both operations at retail Amazon.com and Amazon Web Services. However, a bad flu has caused Jeff to take the week off sick. All database operations on AWS are down. Emotive's client Brands need database operations to be able to a) add an item given as a SKU number to a make-shift database (a data structure of your choice) b) remove a specified item by SKU if one exists or c) check the quantity count and print out 1 if a specified item has count specified otherwise print out 0.

   Our brands give us their request as a command line input of a series of tuples (See example below).

   (1, x) signifies that you add item SKU x to your makeshift database (data structure)
   (2, y) signifies that you remove item SKU y from your makeshift database(data structure) if it exists otherwise do nothing
   (3, z) means that you return 1 if a SKU exists of quantity exactly z in your data structure. Otherwise return 0.

   So for example let us say one of our Brands is UniLever that provide us the following:

6 ←First number just specifies the number of operations you should process
(1, 114)
(1, 115)
(1, 116)
(2, 115)
(3, 2) → Print out 0 (no SKU with frequency count 2 exists)
(3, 1) → Print out 1 since at least 1 SKU with frequency of 1 (Both 114 and 116)

Your function should return [0,1]

3. Bitly is an important service that basically maps and shortens long URLs and replaces them with a different url of the form bit.ly/[Some Hashcode]. Emotive uses Bitly to match URLs to a bitly link that gets sent out to customers and redirects to the longer link.

   The hashcode is usually of the form A-Z and 0-9.

   Assume Bitly is down for a week and you need to architect a system that functions like bitly. Walk through the brief architecture of such a system, talk about the calls, the servers, and the design of the database. Be prepared to talk about scaling such a system to higher levels. How would you handle one url? How would you handle 10 million URLs?