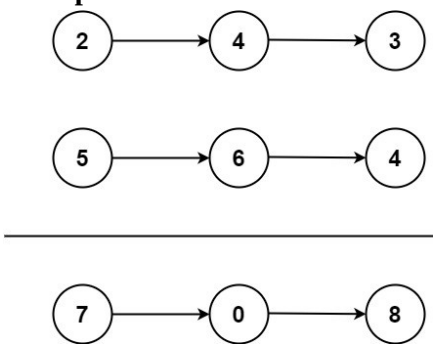
First Challenge:
Implement a Lottery Game
A system that our users can use their chances to win prizes. We have 5 prizes A, B, C, D, and E with different weights 0.1, 0.3, 0.2, 0.15, and 0.25 respectively. Also, we have registered users in our Redis database who are allowed to participate in the lottery only 3 times a day. Suppose the lottery process is a heavy process and takes about 5 seconds.
We want to implement an asynchronous service that accepts the incoming HTTP requests from the clients and performs the lottery, at last, it gives the users the prize as a response.

Request details:
POST *api*/v1/lottery
BODY:
{
UUID: "`f1bc8f04-0500-11ee-be56-0242ac120002`" // UUID of the user
}

Implementation criteria:
1.Layer and directory structure according to the standard of the language you decided to use
2.Use docker compose to run all of the services
3.Writing unit tests is a plus

Second Challenge:
You are given two non-empty linked lists representing two non-negative integers. The digits are stored in reverse order, and each of their nodes contains a single digit. Add the two numbers and return the sum as a linked list.
You may assume the two numbers do not contain any leading zero, except the number 0 itself.

Example 1:




```
Input: l1 = [2,4,3], l2 = [5,6,4]
Output: [7,0,8]
Explanation: 342 + 465 = 807.
```

Example 2:

```
Input: l1 = [0], l2 = [0]
Output: [0]
```

Example 3:

```
Input: l1 = [9,9,9,9,9,9,9], l2 = [9,9,9,9]
Output: [8,9,9,9,0,0,0,1]
```

Constraints:

- The number of nodes in each linked list is in the range [1, 100]
- 0 <= node.value <= 9
- It is guaranteed that the list represents a number that does not have leading zeros.