Queues
(Assignment Questions)

Question 1 : There are n people in a line queuing to buy tickets, where the 0th person is at the front of the line and the (n - 1)th person is at the back of the line.

You are given a 0-indexed integer array tickets of length n where the number of tickets that the ith person would like to buy is tickets[i].

Each person takes exactly 1 second to buy a ticket. A person can only buy 1 ticket at a time and has to go back to the end of the line (which happens instantaneously) in order to buy more tickets. If a person does not have any tickets left to buy, the person will leave the line.

Return the time taken for the person at position k (0-indexed) to finish buying tickets.[Go to Qs]

Examples :
   Input: tickets = [2,3,2], k = 2
   Output: 6
   Explanation: In 1st pass, everyone in the line buys a ticket and the line becomes[1, 2, 1].
   - In 2nd pass, everyone in the line buys a ticket and the line becomes [0, 1, 0].
   The person at position 2 has successfully bought 2 tickets and it took 3 + 3 = 6 seconds.

Question 2 : There are n gas stations along a circular route, where the amount of gas at the ith station is gas[i].

You have a car with an unlimited gas tank and it costs => costs[i] of gas to travel from the ith station to its next (i + 1)th station. You begin the journey with an empty tank at one of the gas stations.

Given two integer arrays gas and cost, return the starting gas station's index if you can travel around the circuit once in the clockwise direction, otherwise return -1. If there exists a solution, it is guaranteed to be unique [Go to Qs]

Examples :

 Input: gas = [1,2,3,4,5], cost = [3,4,5,1,2]

 Output: 3 Explanation:

 Start at station 3 (index 3) and fill up with 4 unit of gas. Your tank = 0 + 4 = 4

 Travel to station 4. Your tank = 4 - 1 + 5 = 8

 Travel to station 0. Your tank = 8 - 2 + 1 = 7

 Travel to station 1. Your tank = 7 - 3 + 2 = 6

 Travel to station 2. Your tank = 6 - 4 + 3 = 5

 Travel to station 3. The cost is 5. Your gas is just enough to travel back to station 3.

 Therefore, return 3 as the starting index.

<mark>Note :</mark> Use Deque to solve the question.

Question 3 : Given an integer K and a queue of integers, we need to reverse the order of the first K elements of the queue, leaving the other elements in the same relative order.

Only following standard operations are allowed on queue.

push(x) : Add an item x to rear of queue

pop() : Remove an item from front of queue

size() : Returns the number of elements in the queue.

front() : Finds front item.

Example :

 Input : Queue is [1, 2, 3, 4, 5] & K = 3

 Output: [3, 2, 1, 4, 5]

 Explanation:

 After reversing the given input from the 3rd position the output will be 3 2 1 4 5.

Bonus Question : Design a data structure that follows the constraints of a Least Recently Used (LRU) cache. Implement the LRUCache class:

LRUCache(int capacity) : Initialize the LRU cache with positive size capacity.
int get(int key) : Return the value of the key if the key exists, otherwise return -1.
void put(int key, int value) : Update the value of the key if the key exists.
Otherwise, add the key-value pair to the cache. If the number of keys exceeds the capacity from this operation, evict the least recently used key.

The functions **get** and **put** must each run in O(1) average time complexity.[Go to Qs]

Note : This question uses an additional data structure, map, that we haven't covered yet. It will be covered in the later chapters. But you can read up about it & try to solve the Qs on your own.

https://telegram.me/+QGUyTripaP4zNTcx

https://telegram.me/+nEKeBr_yhXtmY2Yx

Say Hi to TG for Further Updates:
https://t.me/RepublicDayBot

#TGSFamily