**Fair Waiting Lists**

A viral new Thai noodle restaurant has a huge number of customers piling up in front of their door. The restaurant implemented two systems for customer to be put in the Waiting List: Online Reservation and Walk-In. Usually, the customer who have booked online will be prioritized, however for balancing purpose as the restaurant receive huge load of both online and walk-in, the newly opened restaurant decides to have 2 different waiting lists: One for the Online Reservation and another for Walk-In. The customer will be served alternately between the 2 waiting lists with **First Come First Serve** basis. Because the long waiting time (2-3 hours), there are also customers who choose to cancel.

Given the number of Online Reservation customers , Walk-In customers , and cancelling customers you will be asked for a unique integer id for each online customer (), walk-in customer (), and cancelling customer (). The task is to list out the serving order until all the customers are served.

**Input:**

1. The first line of input will use format:

where is an integer for number of Online Reservation Customers and is an integer for number of Walk-In Customers.

1. The second line of input will use format:

where each is an integer for a unique id of an Online Reservation customer. Assume that the customer ids are always unique so no need to validate the id.

1. The third line of input will use format:

Where each is an integer for a unique id of a Walk-In customer. Assume that the customer ids are always unique so no need to validate the id.

1. The fourth line of input will use format:

Where is the number of customers who cancelled. Assume that this number always has maximum value of (no case where all customers cancelled), so no need to validate whether the number is less than or not.

1. The fifth line of input will use format:

Where each is an integer for a unique id of the customer who cancelled. Assume that each id inputted always exist, so no need to validate whether the id exist or not.

**Output:**

List out the id of the customers in the order of serving until all customers are served, starting from the Online Reservation customer. The customer who cancelled will be gone from this list.

**Example**:

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 2  10 15  20 13  0 | 10 20 15 13 |
| **Explanation:**   1. First, we have 2 Online customers and 2 Walk-in customers. 2. 10 and 15 are inputted as the id of online customers, then 20 and 13 are inputted as the id of walk-in customers. 3. With both of those id, we can set the order alternately starting from online customer which resulted in 10, 20, 15, 13. 4. There are 0 customers which cancelled, so the program will skip the fifth input line and print the output. | |
| 3 5  1 5 3  9 8 10 19 18  2  5 10 | 1 9 8 3 19 18 |
| **Explanation:**   1. First, we have 3 Online customers and 5 Walk-in customers. 2. 1, 5, and 3 are inputted as the id of online customers, then 9, 8, 10, 19, 18 are inputted as the id of walk-in customers. 3. With both of those id, we can set the order alternately starting from online customer which resulted in 1, 9, 5, 8, 3, 10, 19, and 18. Because there are no more walk-in customer after 10, customer id 19 and 18 can be served. 4. However, there are 2 customers which cancelled with id 5 and 10, so we delete the id from the previous order to become 1, 9, 8, 3, 19, and 18. | |
| 6 3  10 8 9 20 19 16  34 27 11  3  10 16 9 | 34 8 27 11 20 19 |
| **Explanation:**   1. First, we have 6 Online customers and 3 Walk-in customers. 2. 10, 8, 9, 20, 19 and 16 are inputted as the id of online customers, then 34, 27 and 11 are inputted as the id of walk-in customers. 3. With both of those id, we can set the order alternately starting from online customer which resulted in 10, 34, 8, 27, 9, 11, 20, 19, and 16. Because there are no more online customer after 20, customer id 19 and 16 can be served. 4. However, there are 3 customers which cancelled with id 10, 16 and 9, so we delete the id from the previous order to become 34, 8, 27, 11, 20, and 19. | |

Your code must use **Double Linked List** data structures to:

1. **Enqueue** (Insert) the customer Ids into **two separate queues**.
2. **Enqueue** the Ids from two queues above **alternatingly** into a **new queue**.
3. **Delete** specific values from the **new queue**.
4. Print out the value in the new queue.