**Batch: B3 Roll. No.: 121**

**Experiment: 12**

**Grade: AA / AB / BB / BC / CC / CD /DD**

|  |
| --- |
| **Title:**  Using virtual labs to understand the data structures |

**Objective:** Use of virtual labs to understand the concepts and theory with examples and verify the same with practice questions.

**Expected Outcome of Experiment:**

|  |  |
| --- | --- |
| **CO** | **Outcome** |
| **CO1** | Explain the different data structures used in problem solving |
| **CO2** | Apply linear and non-linear data structure in application development |
| **CO3** | Demonstrate sorting and searching methods. |

**Websites/books referred:**

1. [**https://ds1-iiith.vlabs.ac.in/exp/stacks-queues/index.html**](https://ds1-iiith.vlabs.ac.in/exp/stacks-queues/index.html)
2. [**https://ds1-iiith.vlabs.ac.in/exp/stacks-queues/queues/index.html**](https://ds1-iiith.vlabs.ac.in/exp/stacks-queues/queues/index.html)
3. [**https://ds1-iiith.vlabs.ac.in/exp/stacks-queues/queues/queues-quiz.html**](https://ds1-iiith.vlabs.ac.in/exp/stacks-queues/queues/queues-quiz.html)

Abstract: the virtual lab experiments help in understanding how various data structures work. They also emphasize on some important applications of various data structures and enable students to get familiarized with how certain applications can benefit from the choice of data structures.

Assigned data structure: *(Teacher would assign one of the following to one student)*

1. Stack - <https://ds1-iiith.vlabs.ac.in/exp/stacks-queues/stacks/stackdemo.html>
2. Infix and postfix - https://ds1-iiith.vlabs.ac.in/exp/infix-postfix/evaluation-of-postfix-expressions/postfix\_eval.html
3. Queue - <https://ds1-iiith.vlabs.ac.in/exp/stacks-queues/stacks/stackdemo.html>
4. Bubble sort - <https://ds1-iiith.vlabs.ac.in/exp/bubble-sort/bubble-sort/bsexercise.html>
5. Graph DFS - <https://ds1-iiith.vlabs.ac.in/exp/depth-first-search/index.html>
6. Graph BFS - <https://ds1-iiith.vlabs.ac.in/exp/breadth-first-search/index.html>
7. Binary search tree - <https://ds1-iiith.vlabs.ac.in/exp/binary-search-trees/bst-insert/bstInsert.html>
8. Hash tables - <https://ds1-iiith.vlabs.ac.in/exp/hash-tables/quadratic-probing/qp_practice.html>
9. Linked list - https://ds1-iiith.vlabs.ac.in/exp/linked-list/singly-linked-list/sllpractice.html

**Aim / learning objective of the assigned expt:**

The learning objectives of the Queue Experiment are as follows:

1. Gain the concept of Queues.
2. Understand the basic operations of Queues.
3. Practice the operations of Queues.
4. Test the conceptual understanding with a quiz.

**Concept and algorithm of the application/activity followed:**

**What are Queues?**

* Imagine a line of people waiting at a ticket counter. People are served in the order they come, that is, people who come first are served first whereas people who come later are served after that.
* Let the action of someone joining a queue be called enqueue and someone being serve and getting out of the queue be called dequeue.
* A type of structure, similar to the example of the line of people, can be represented as a data structure. Such a data structure is known as a queue.

Types of Queues

SIMPLE QUEUE: A simple queue is a type of queue where insertion is at the end of the queue and removal is at the front.

CIRCULAR QUEUE : A circular queue is a type of queue where the last element is connected to the first. An element is added to the end of the queue and removed from the front.

PRIORITY QUEUE : A priority queue is a type of queue where the elements are arranged in some priority order. An element with the highest priority is removed first and insertion occurs according to the priority.

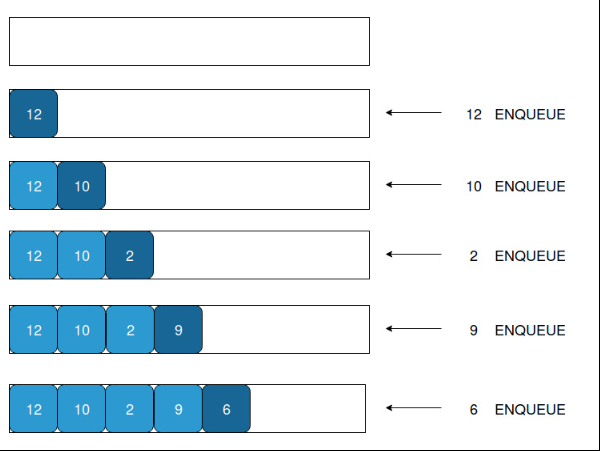
DOUBLE ENDED QUEUE: A double ended queue is a type of queue where insertion and deletion can happen at both ends of the queue.

Queue Operations and ApplicationsJust like how we saw in the example of the queue of people, a queue data structure has two types of operations : enqueue and dequeue. As we can see, a queue is an example of a first in, first out data structure (FIFO). That is, an element that is enqueued first is the first to be dequeued.Enqueue operation happens at "REAR" pointer whereas dequeue operation happens at "FRONT" pointer.

Queues have a variety of applications. Lets explore a few of them. -> Processing requests in a website : When a website is visited by a large number of users, it puts the users in a queue - users that vists the website first are processed earlier. -> Priority Queue in Operating Systems : An operating system has many tasks to perform and tasks are ordered according to the most important ones. This makes use of multilevel priority queues.

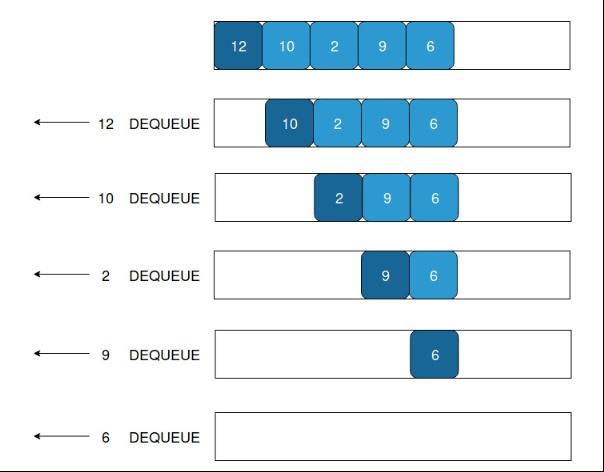
Enqueue Operation

Example: 12, 10, 2, 9, 6



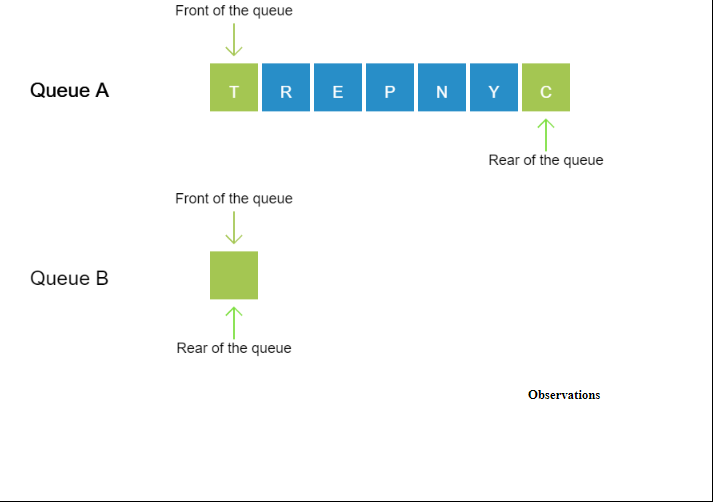
Dequeue Operation:

Example: 12, 10, 2, 9, 6

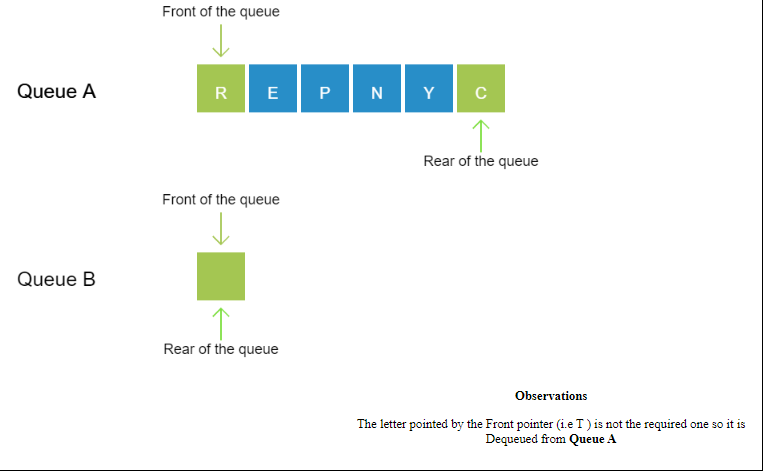


**Demo execution screenshots:**

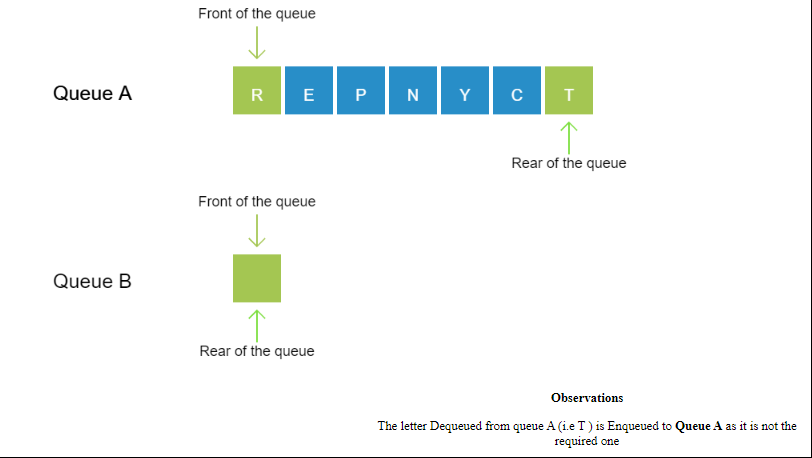
**Step 1:**



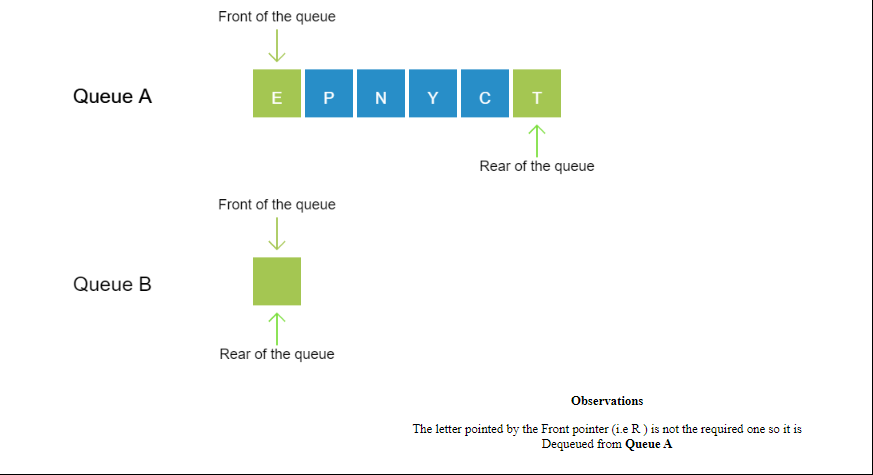
**Step 2:**



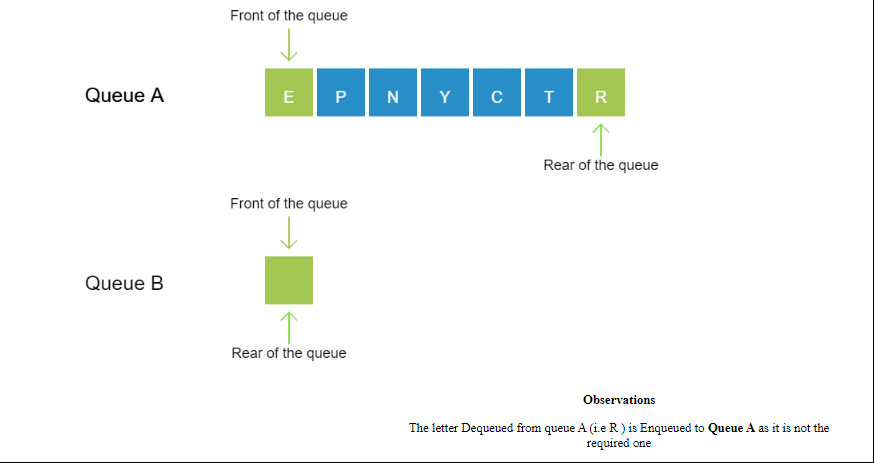
**Step 3:**



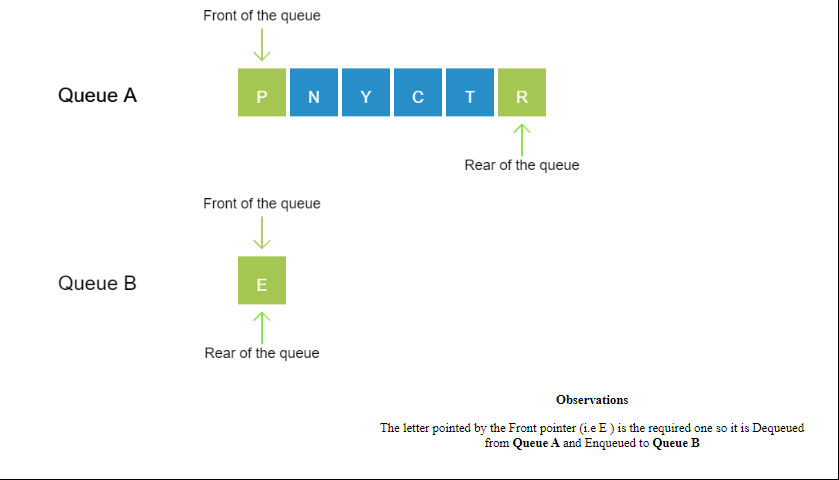
**Step 4:**



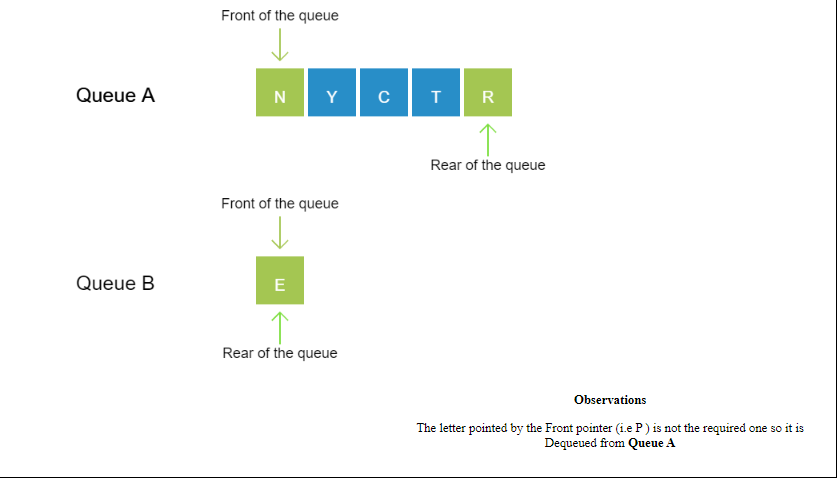
**Step 5:**



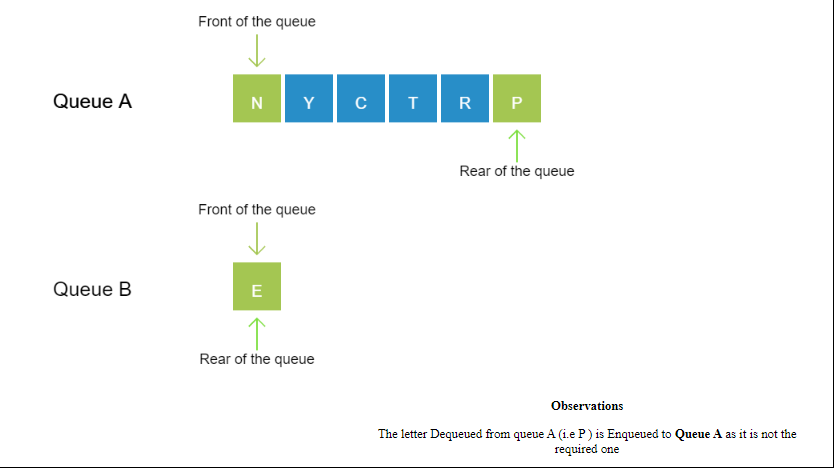
**Step 6:**



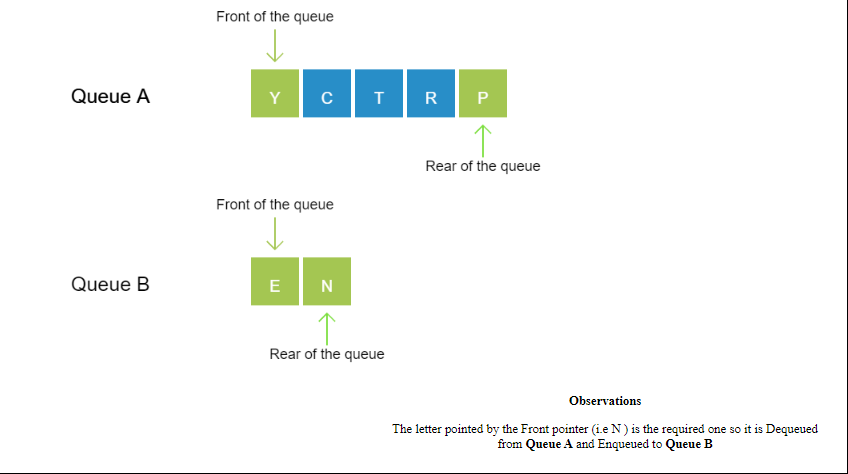
**Step 7:**



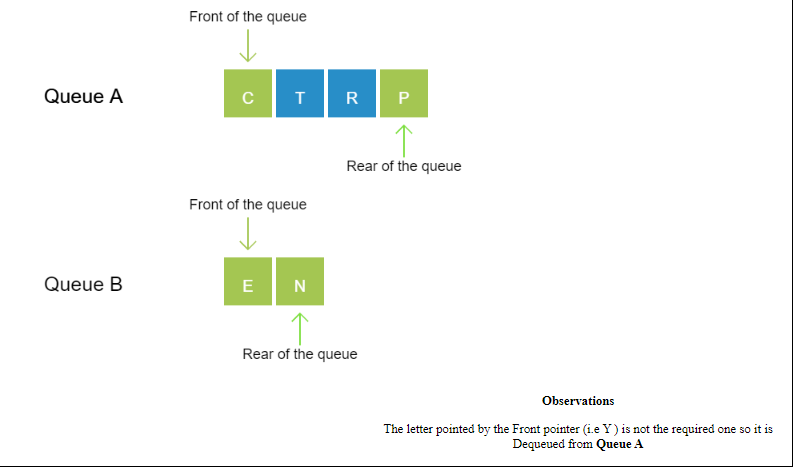
**Step 8:**



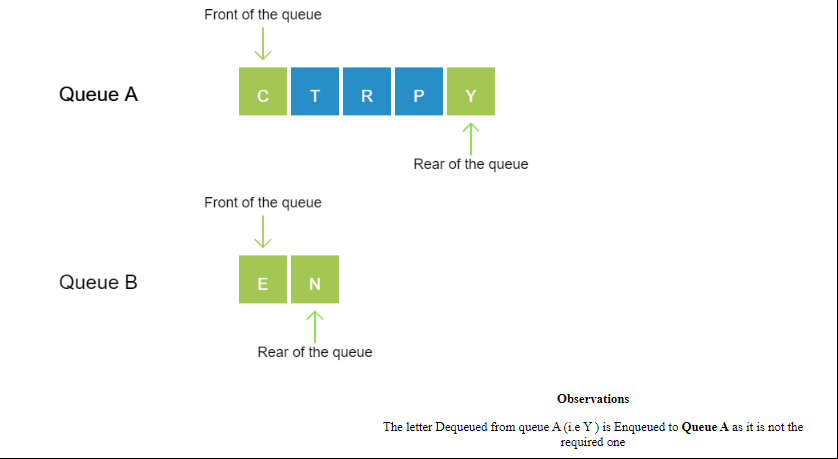
**Step 9:**



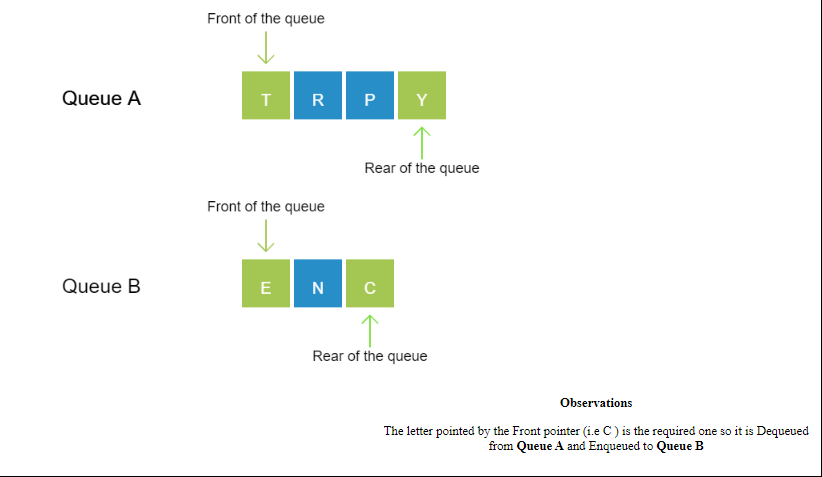
**Step 10:**



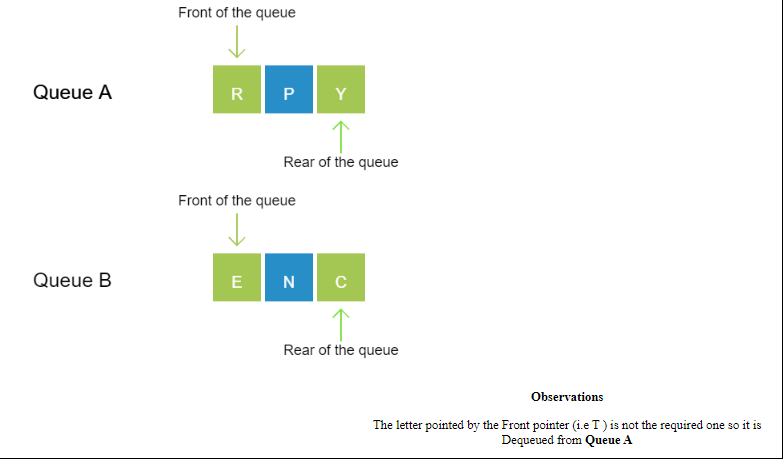
**Step 11:**



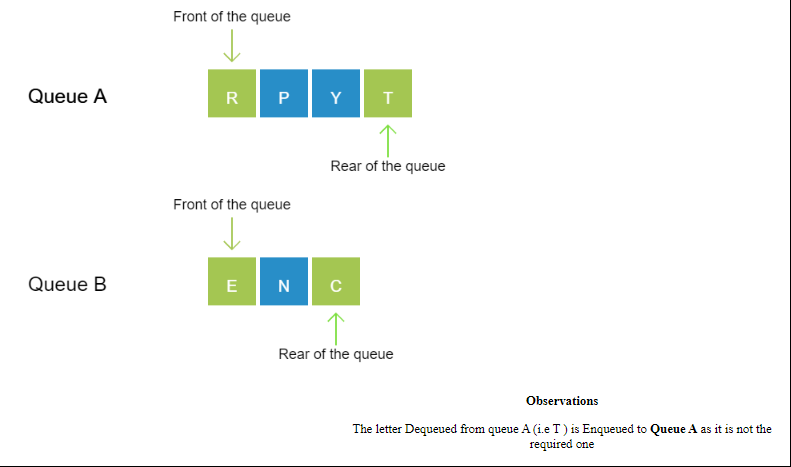
**Step 12:**



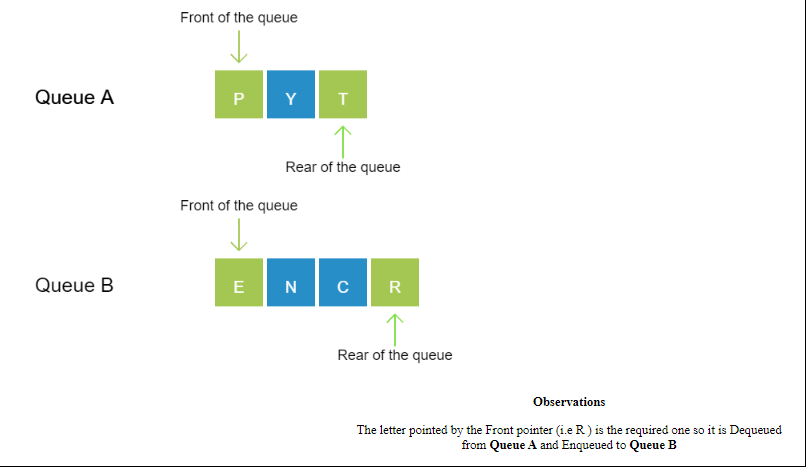
**Step 13:**



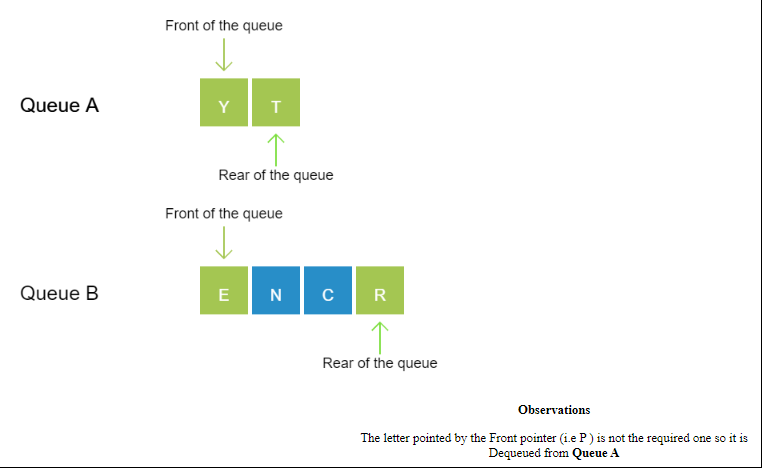
**Step 14:**



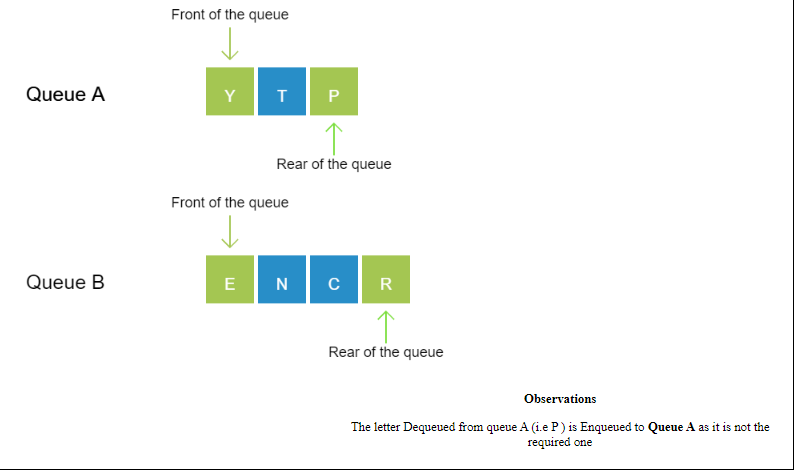
**Step 15:**



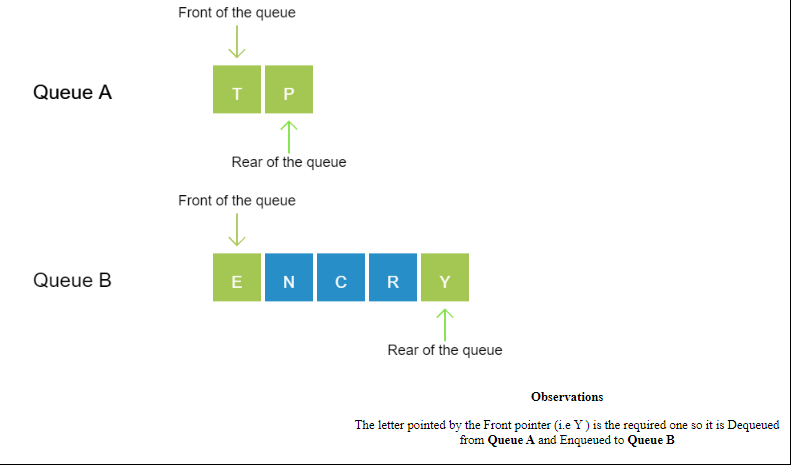
**Step 16:**



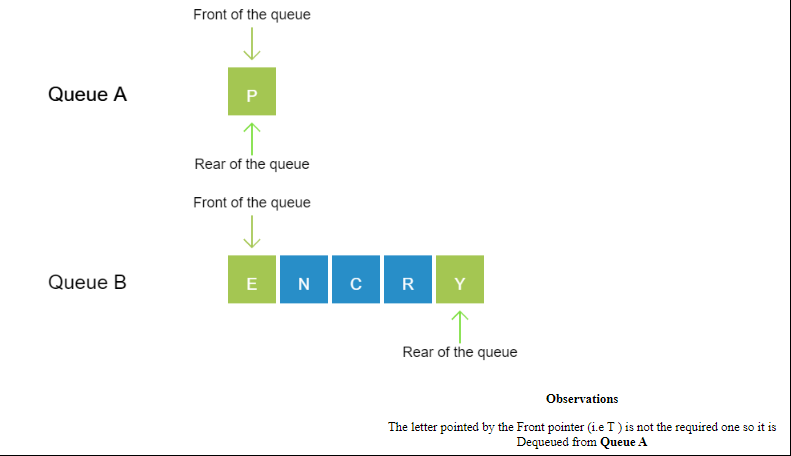
**Step 17:**



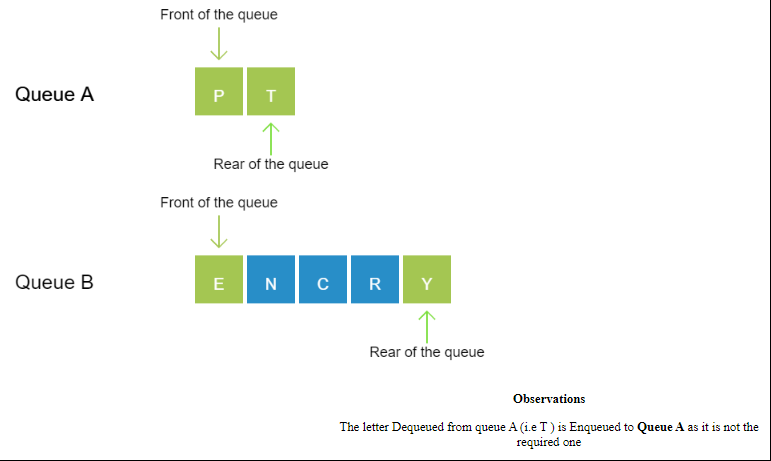
**Step 18:**



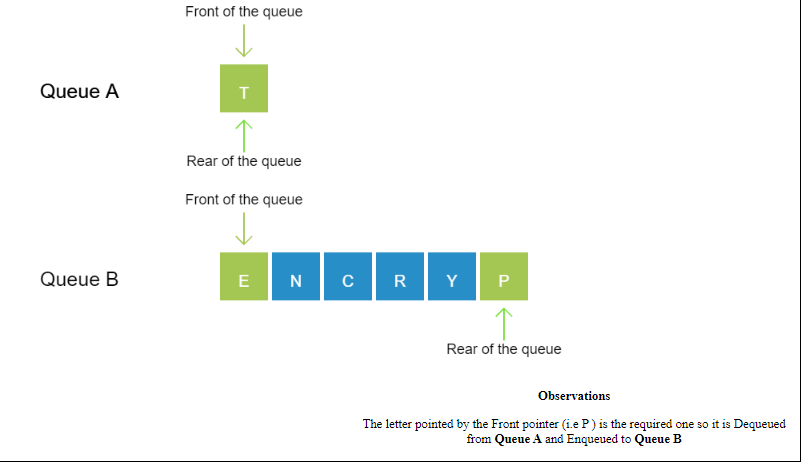
**Step 19:**



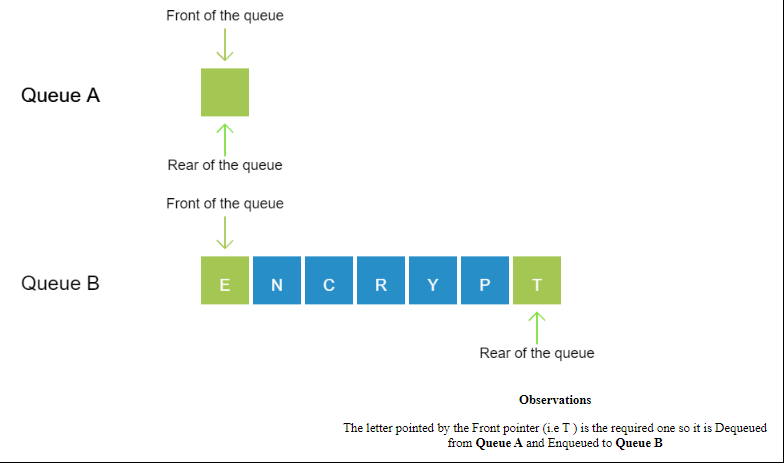
**Step 20:**



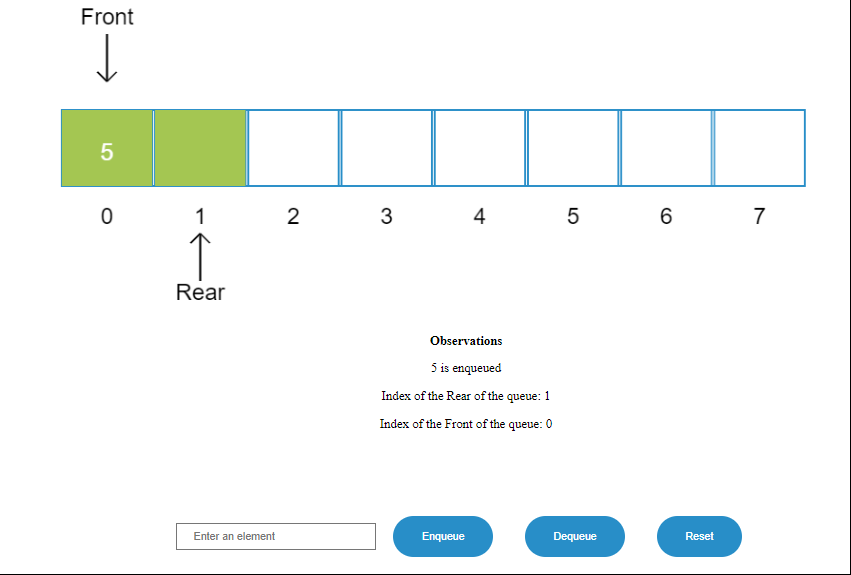
**Step 21:**

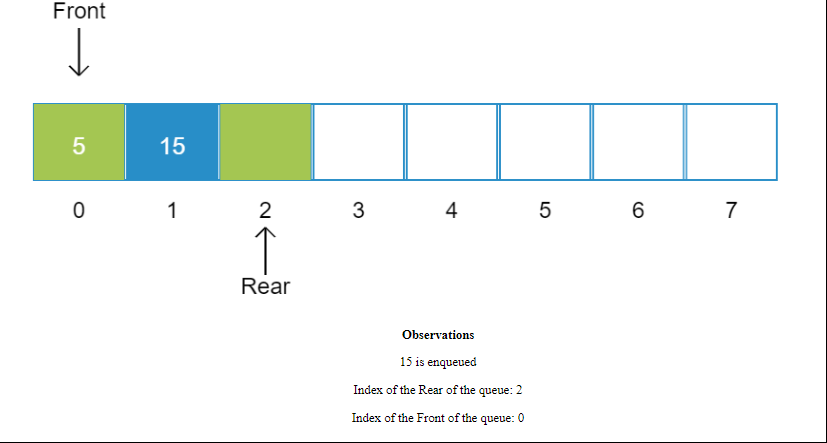
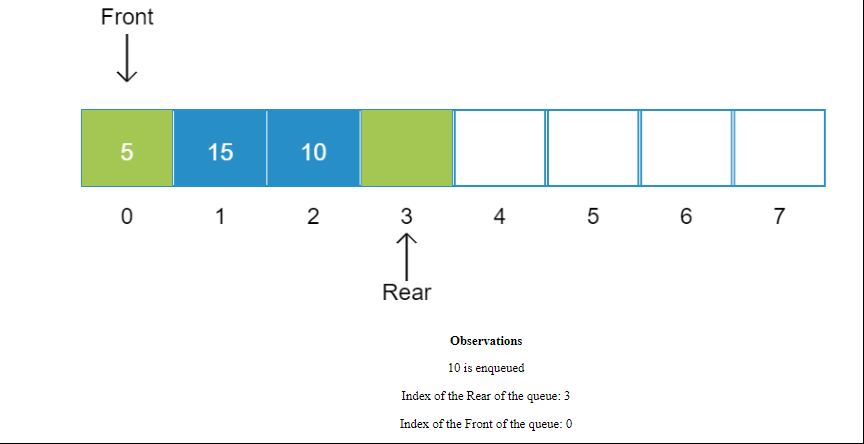
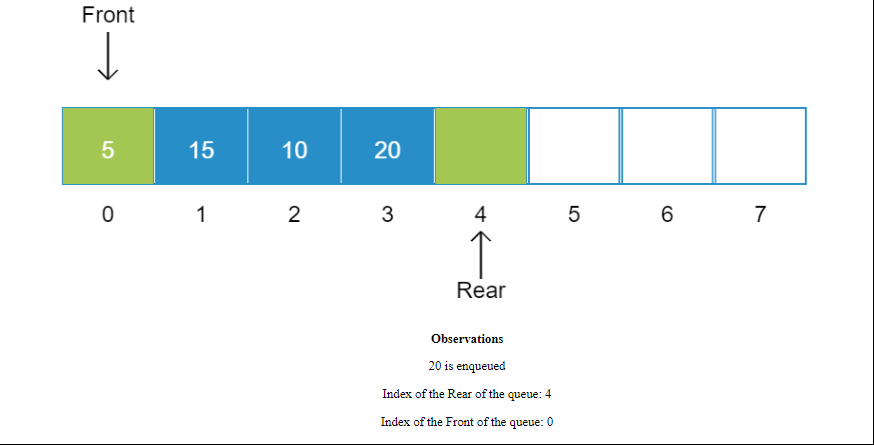
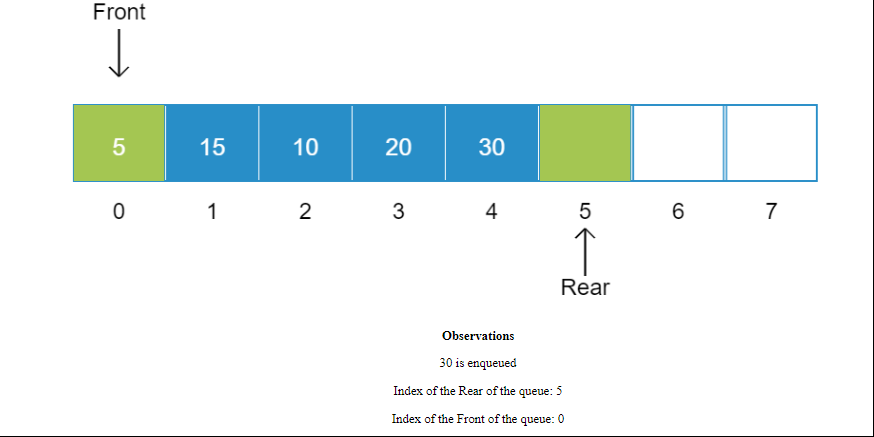
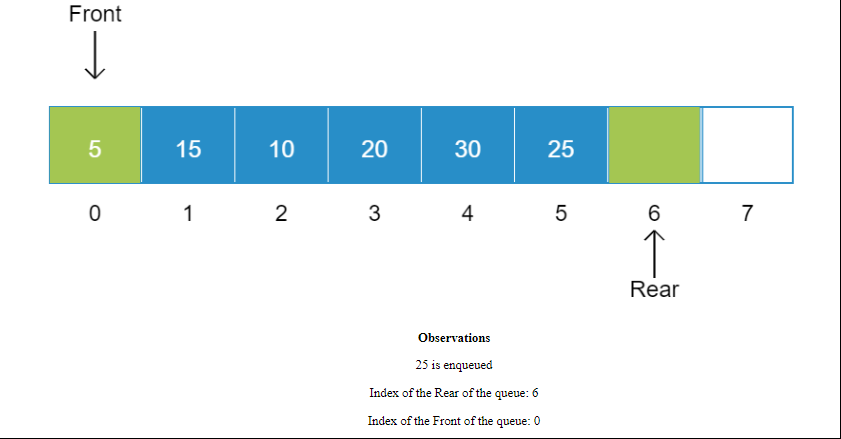
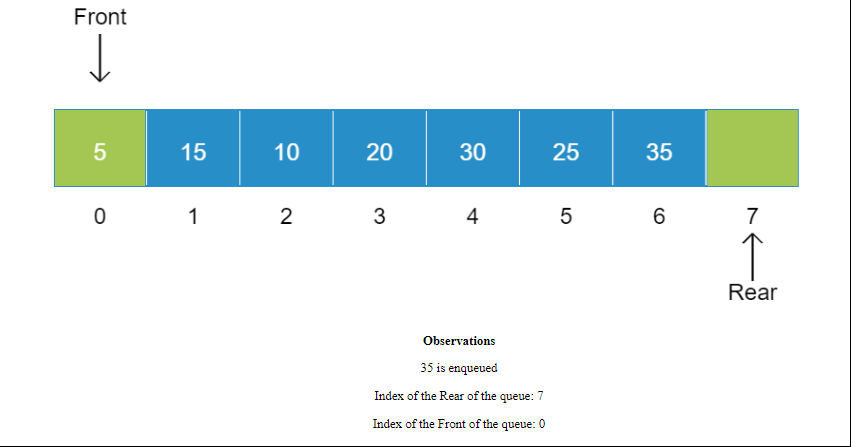
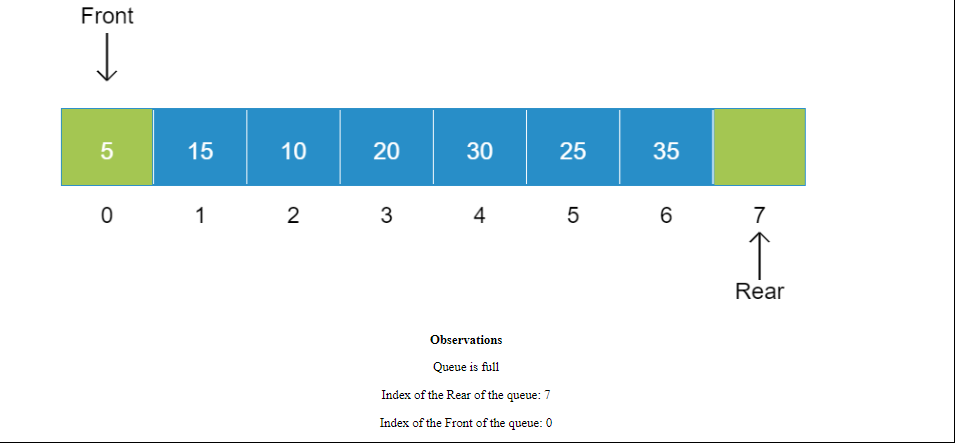


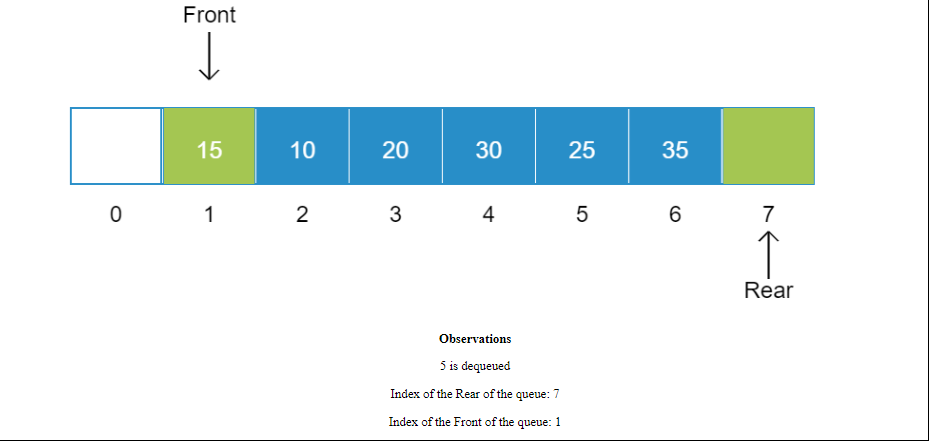
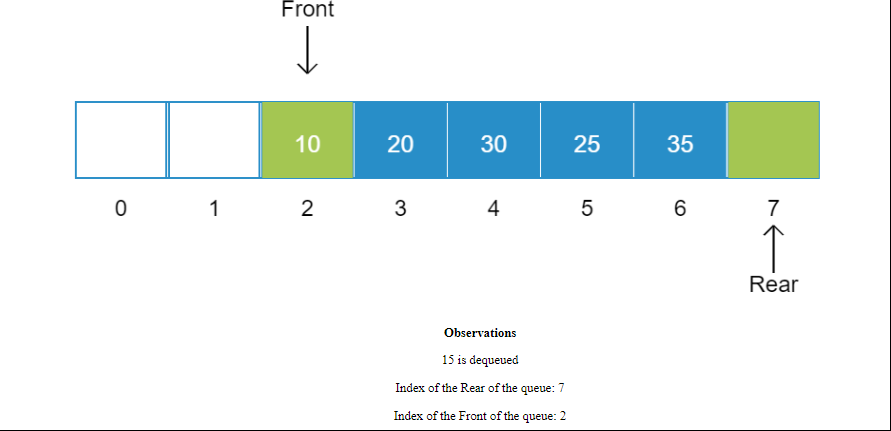
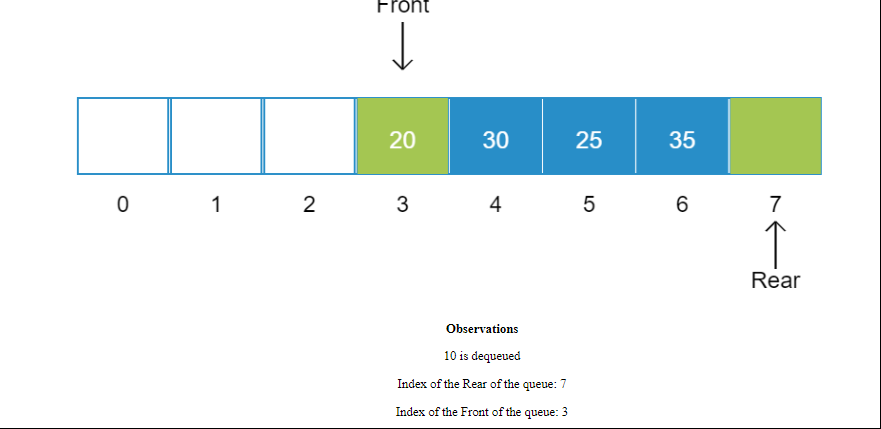
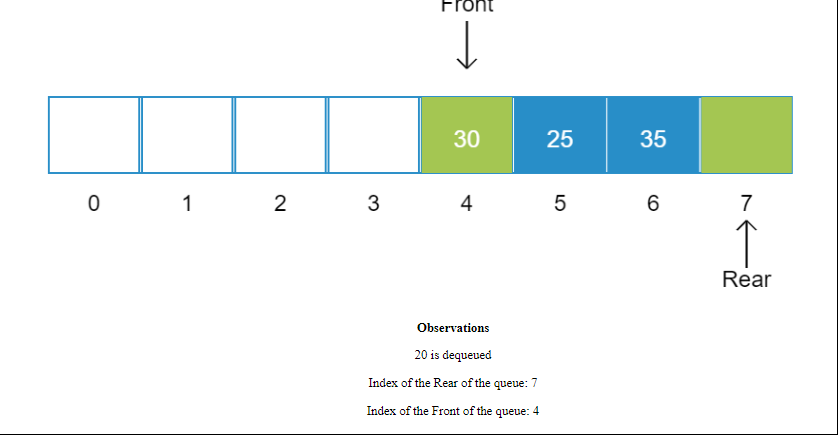
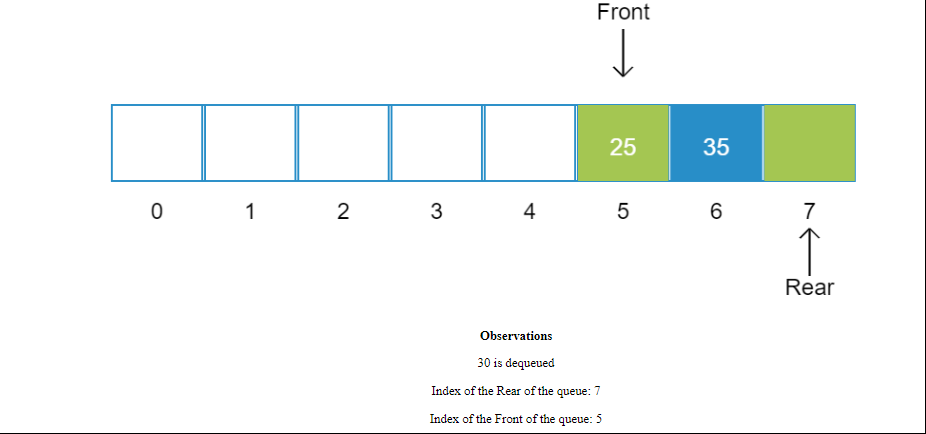
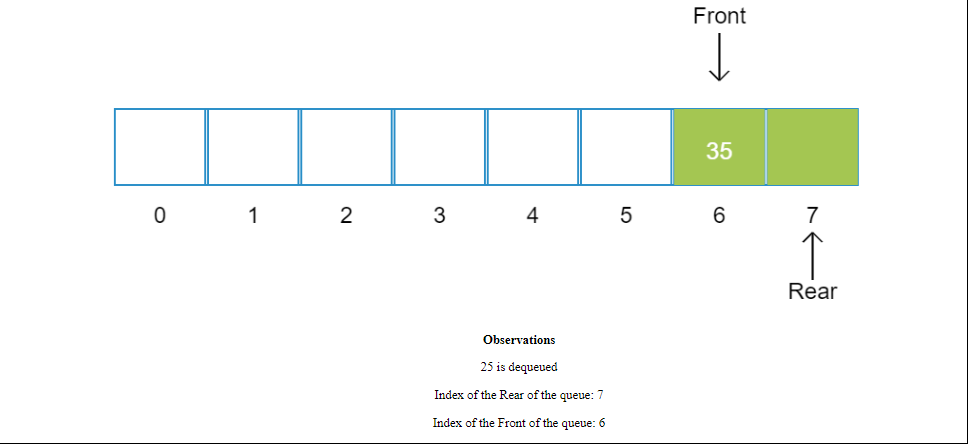
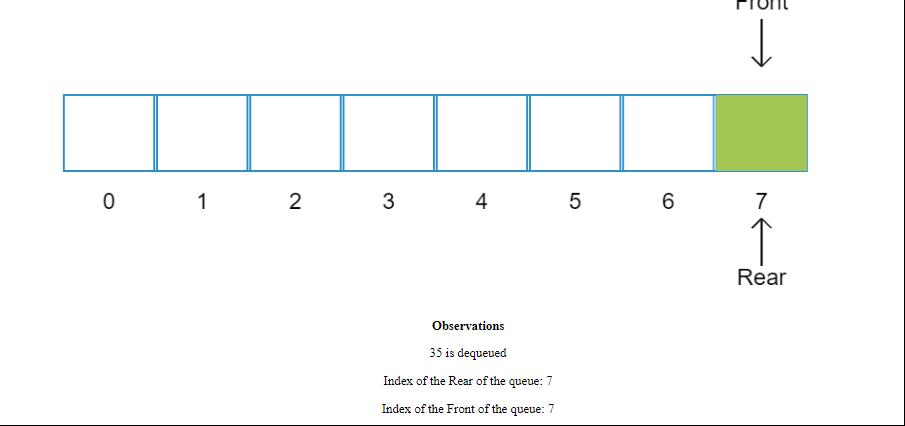
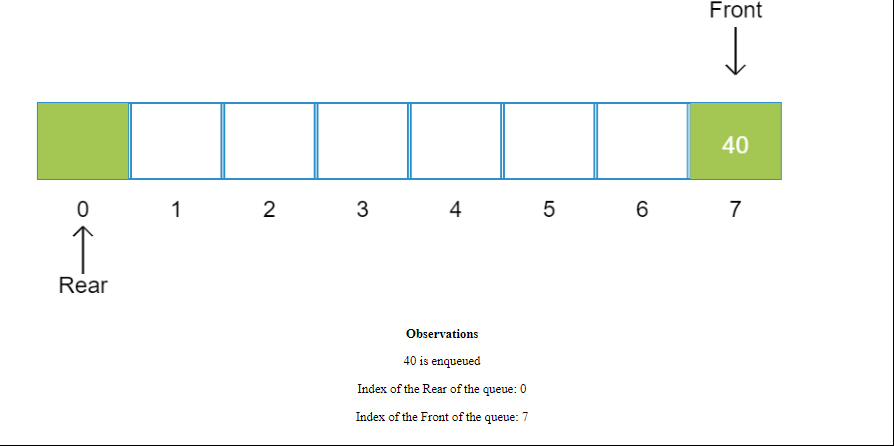
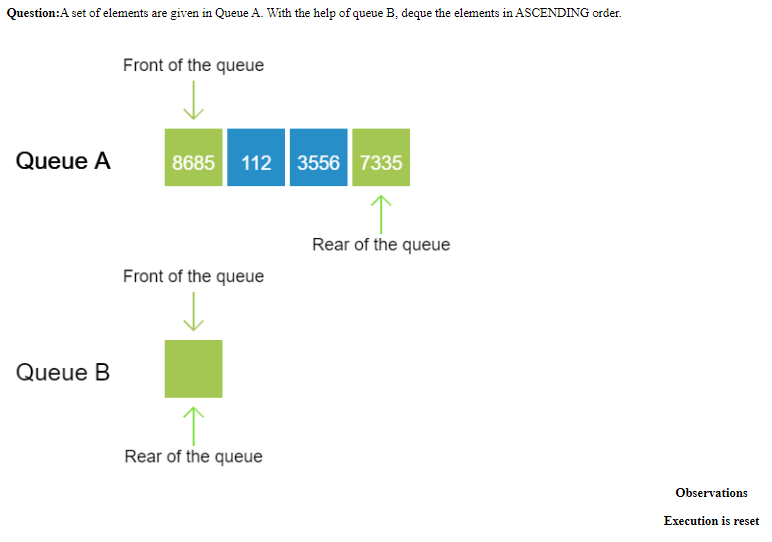
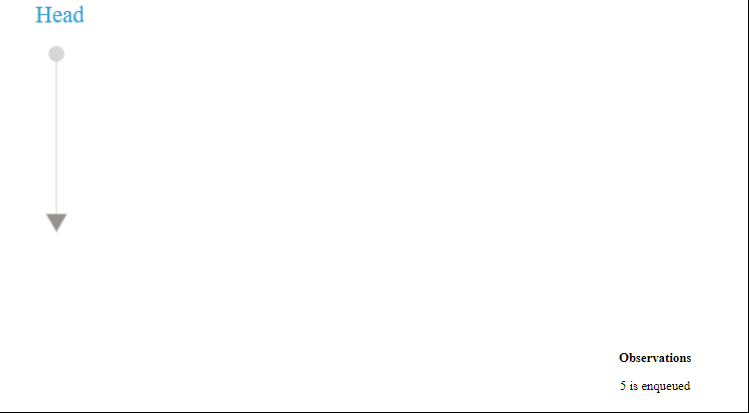
**Step 22:**

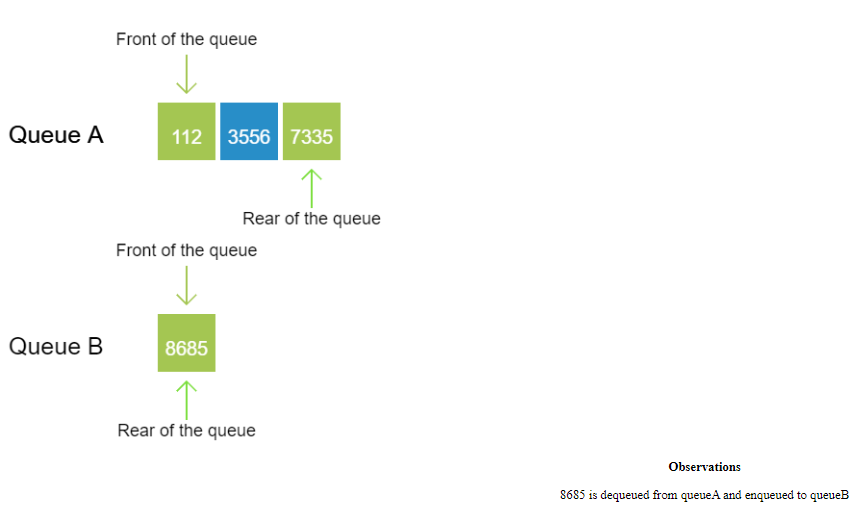


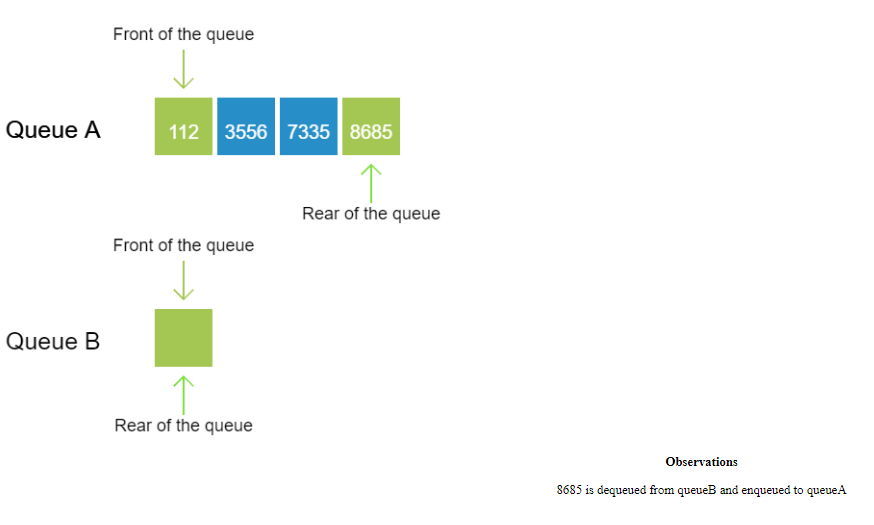
**Practice problem screenshots:**

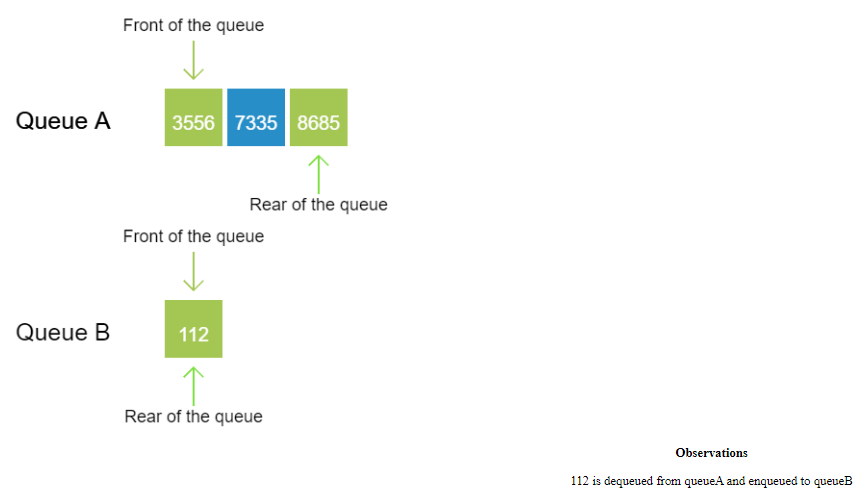


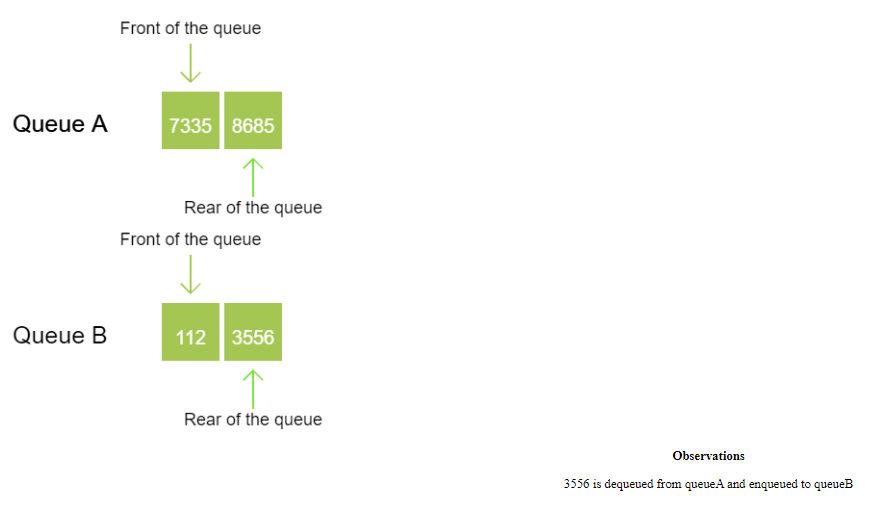
      

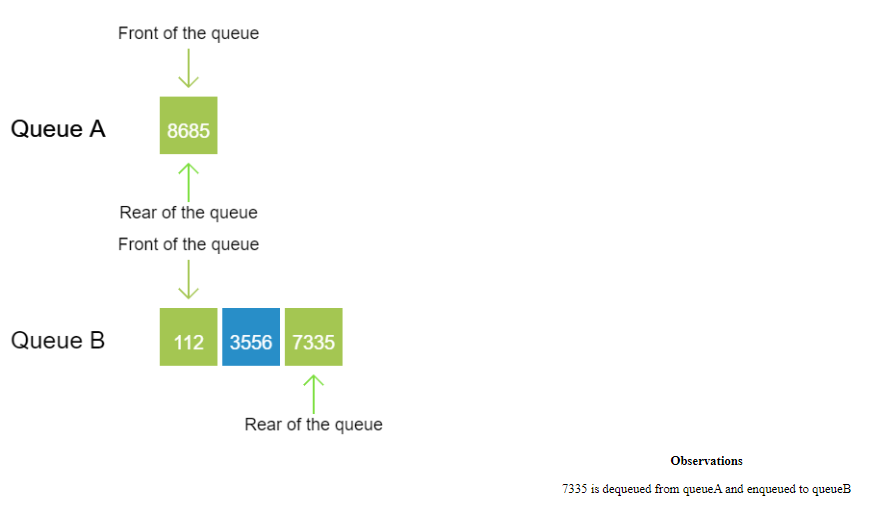
        

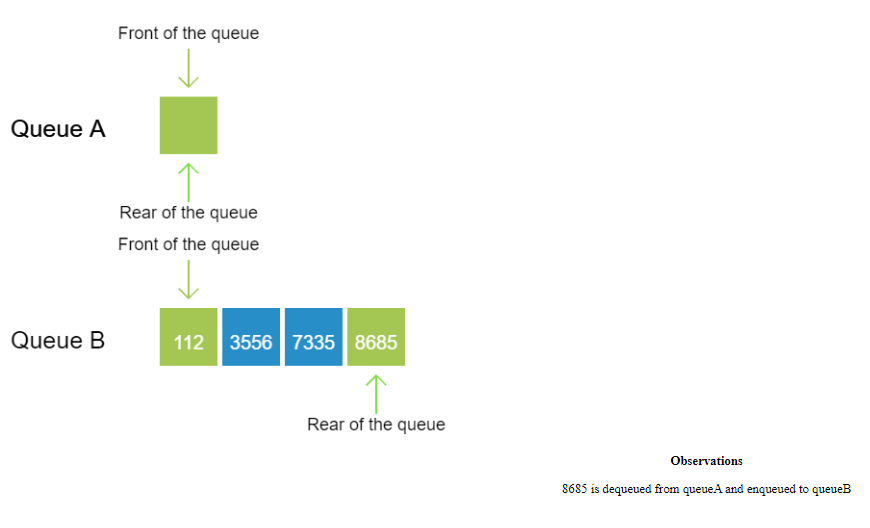
****

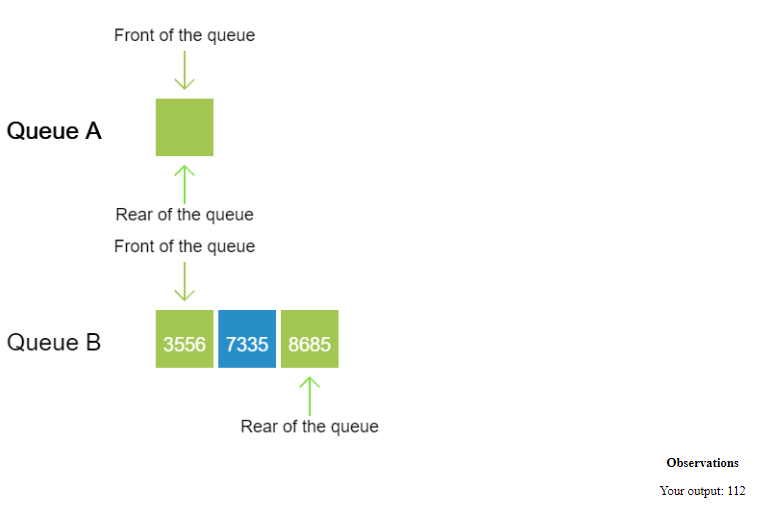
****

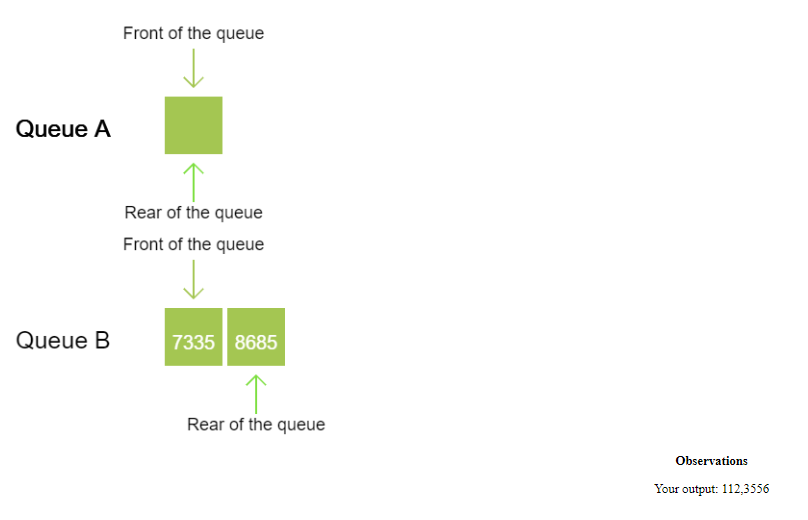
****

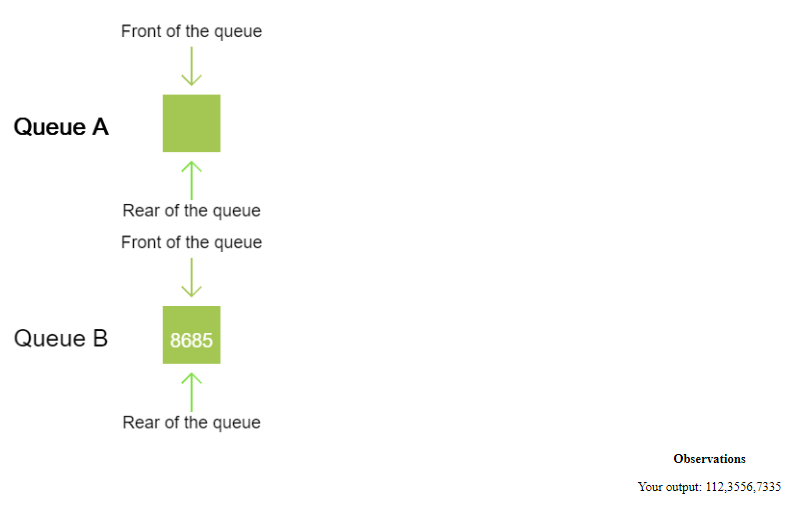
****

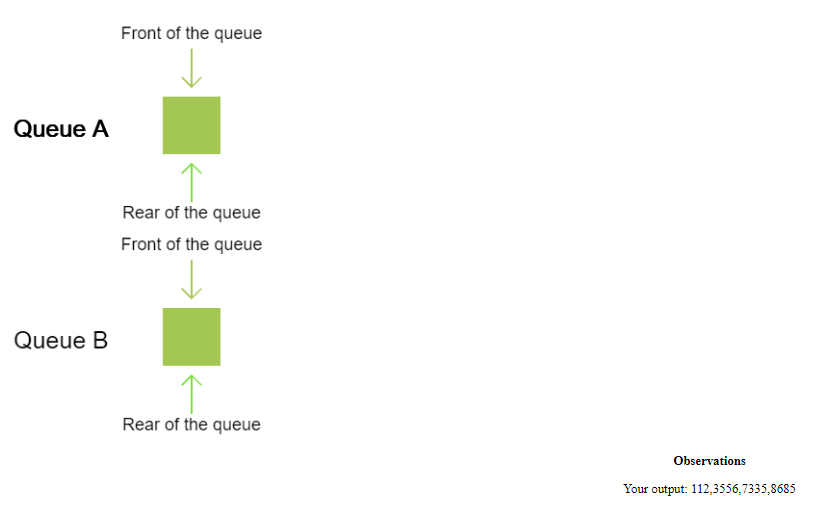
****

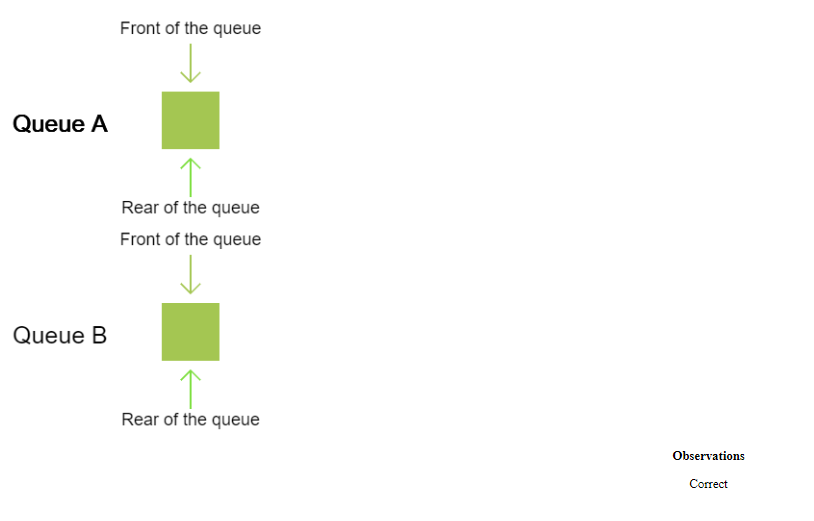
****

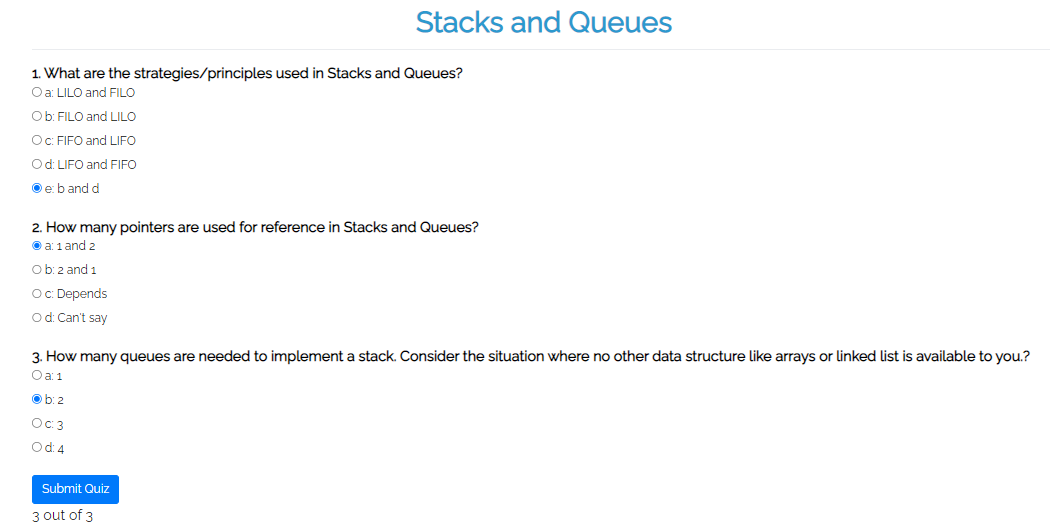
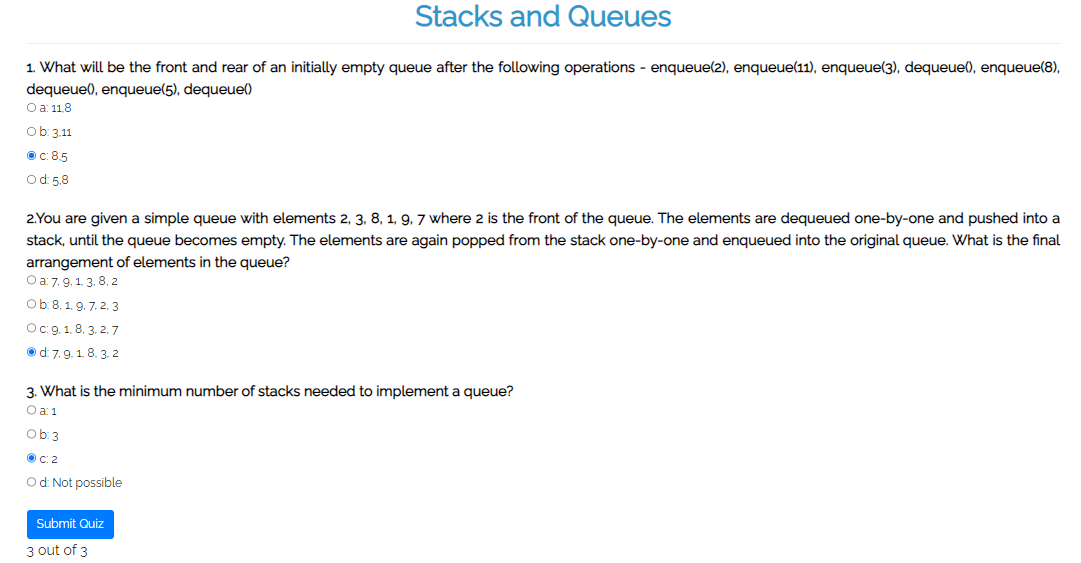
****

****

****

****

****

**Quiz screenshots**

**Conclusion and your take away after performing the virtual lab experiment: -**

Thus, in this experiment, the concept of stack and queue has been learnt in a practical manner. The take away is that queue data structure is useful in a waiting-line situation, where the process which was first admitted should be executed first (i.e., First-In-First-Out). It can also be used for applications like network security (using the concept of encryption, which has been shown in the example above with “ENCRYPT” as the data).