Lab 6: Queue

CLO: 01,02,04

Queue:

A Queue is a linear structure which follows a particular order in which the operations are performed. The order is **First In First Out (FIFO).** A good example of a queue is any queue of consumers for a resource where the consumer that came first is served first. The difference between stacks and queues is in removing. In a stack we remove the item the most recently added; in a queue, we remove the item the least recently added.

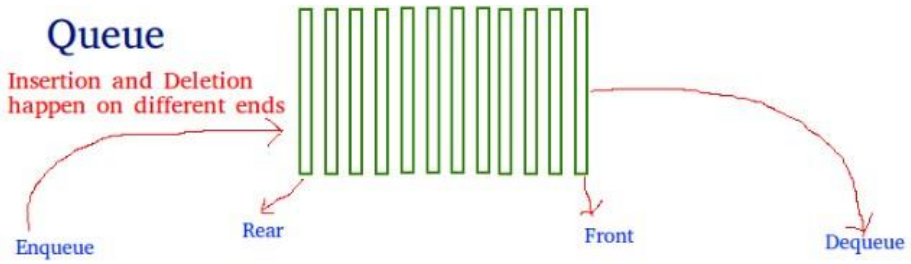

Figure 1. Queue data structure (FIFO)

Queue operations may involve initializing or defining the queue, utilizing it, and then completely erasing it from the memory. Some of the operations are:

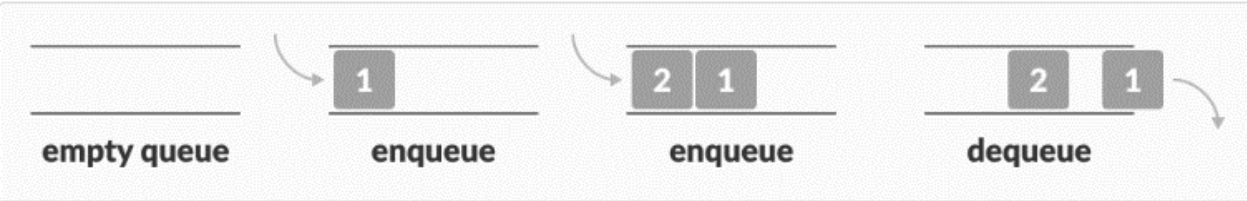

Figure 2. Queue data structure (FIFO)

- enqueue() − add (store) an item to the queue.
- dequeue() − remove (access) an item from the queue.
- isfull() − Checks if the queue is full.
- isempty() − Checks if the queue is empty.

Lab Task

Task 1:

Implement following functions for array-based queue:

a) Function Enqueue to add element to queue.

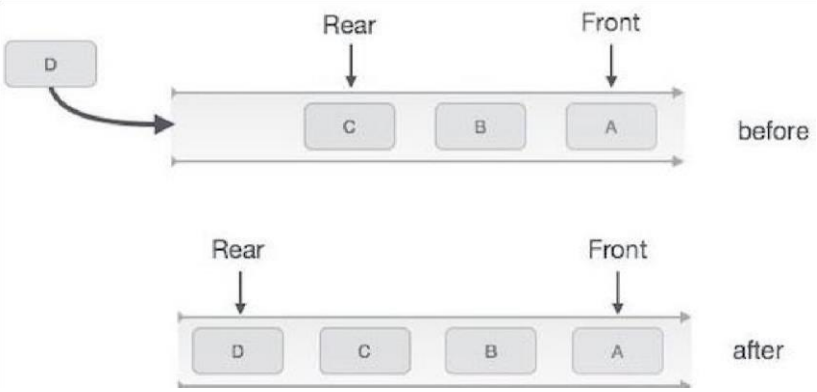

Figure 3. Queue Enqueue

b) Function Dequeue to remove element from queue.

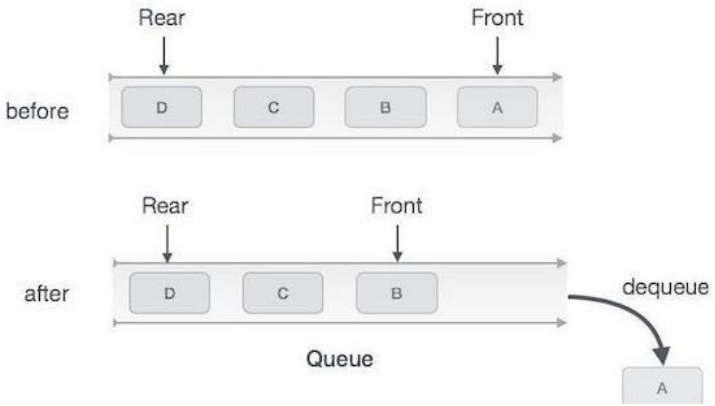

Figure 4. Queue Enqueue

c) Function **Display** to print data of queue.
d) Function **isfull()** to if the queue is full.
e) Function **isempty()** to check if the queue is empty.

Task 2:

Implement following functions for link list-based queue:

a) Function **Enqueue** to add element to queue.
b) Function **Dequeue** to remove element from queue.
c) Function **Display** to print data of queue.
d) Function **isempty()** to check if the queue is empty.

Task 3:

Implement **Priority Queue** functions for link list-based queue:

:

a) **insert(value, priority**): Add an element with a given value and priority to the priority list. Lower priority values indicate higher priority.
b) **getHighestPriority():** Retrieve and remove the element with the highest priority from the priority list. If multiple elements have the same priority, retrieve the one that was inserted first (FIFO).
c) **isEmpty():** Check if the priority list is empty.

Your implementation should ensure that elements are prioritized according to their assigned priority values.