

# Queue

## Additional Document

---

### Queue

This is another type of linear data structure that you can use in various situations. This data structure follows the FIFO (First In First Out). A real-life example of this data structure you can observe in process scheduling where all the tasks which need to be executed will be aligned based on their property and will be executed one by one. You will be using an array and Linked List to implement this data structure.

You can consider stack as people waiting at the ticket counter. Those who came first will be served first. So more people can come and stand in the line from one end and another end will be used to serve people.

In this segment, you will be working on the following algorithms.

1. **Initialization and declaration:** This problem statement will work on the initialization and declaration part of the Queue. Here we will insert a few elements in the queue and then print them accordingly.
2. **Enqueue:** this terminology is used in the queue to showcase insert operation. In the queue, we have two variables: front and rear. The rear will be used to insert the new node and elements.
3. **Dequeue:** This terminology is used to showcase delete operation. In this algorithm, "front" variable will point to the element which can be deleted.
4. **Isempty:** This algorithm will check if the given queue currently holds any element into it. If it is empty it will return True else False.