



DOPE (/blog/)

Queue in C

🕒 May 26, 2017 🔖 C (/blog/tag/c/?tag=c) QUEUE (/blog/tag/queue/?tag=queue) DATA STRUCTURE (/blog/tag/data-structure/?tag=data-structure) 👁 6914



Become an Author

[\(/blog/submit-article/\)](/blog/submit-article/)

Download Our App.



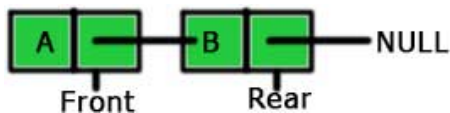
(<https://play.google.com/store/apps/details?id=com.blogsdope&pcampaignid=MKT-Other-global-all-co-prtnr-py-PartBadge-Mar2515-1>)

A queue in computer science is very much similar to the queue in our day to day life. In a queue, we add any item at the rear of the queue and remove any item from the front of the queue. In computer science also, insertions occur at the rear and deletions at the front of the queue. We put the newest element at the rear of the queue and while removing, we remove the element at the front of the queue i.e., First-in, First-out (FIFO). The picture below represents a queue.



Thus, a queue is an abstract data type where new items are inserted at the end and removed from the front. There are some terms and operations we use with a queue. Let's go through them one by one.

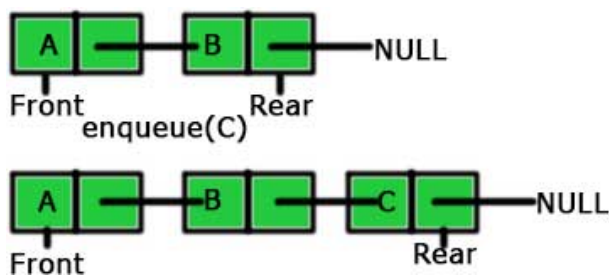
The rear of a queue always points to NULL.



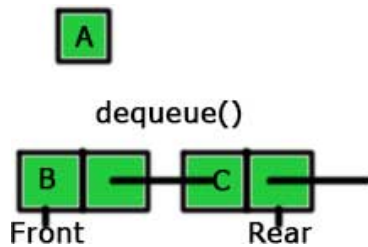
You can see that the rear of the queue in the above picture is pointing to NULL.

Two major operations possible on a queue are enqueue and dequeue. Let's go through them.

- ***enqueue*** → 'enqueue' is a function in a queue which adds a new element in the queue. As a new element is added at the rear of a queue, so 'enqueue' adds a new node at the rear of the queue.



- ***dequeue*** → 'dequeue' is opposite of enqueue. It returns and removes a node from the front of a queue.



In the picture shown above, 'dequeue' returned 'A' and removed that node from the previous queue.

There are also some other operations which we can use on a queue. They are listed below.

- ***initialize()*** → To initialize a queue
- ***front()*** → To return the element at the front of the queue.
- ***isempty()*** → To check whether the queue is empty or not.
- ***isfull()*** → To check whether the queue is full or not.

This article was just about introducing you to the queue. Next two articles are on coding up a queue using linked list and array.

Next:

- Making a queue using linked list in C
(<https://www.codesdope.com/blog/article/making-a-queue-using-linked-list-in-c/>)
- Making a queue using an array in C
(<https://www.codesdope.com/blog/article/making-a-queue-using-an-array-in-c/>)

Liked the post?



(<https://www.facebook.com/sharer/sharer.php?u=https://www.codesdope.com/blog/article/queue-in-c/>)



(<https://twitter.com/intent/tweet?url=https://www.codesdope.com/blog/article/queue-in-c/&text=Queue in C>)