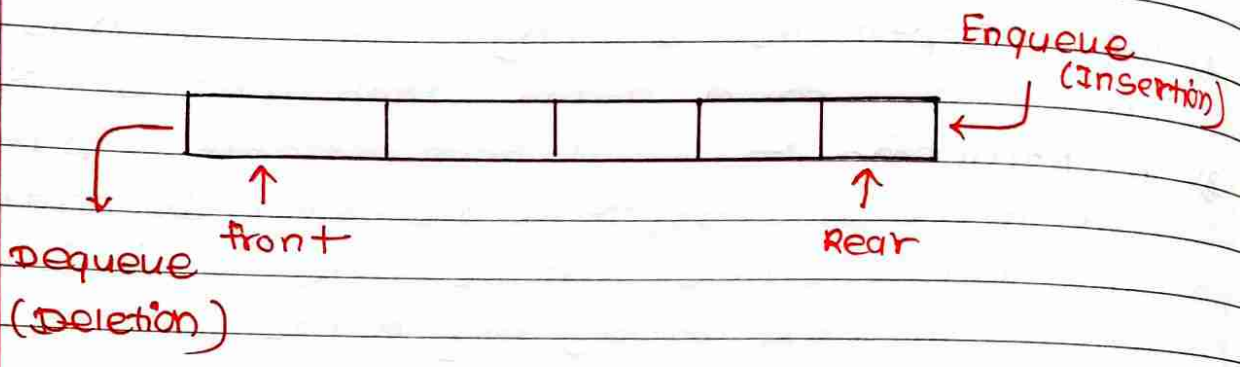
Queue :- A queue can be defined as ordered list which enables insert operations to be performed at one end called REAR and delete operations to be performed at another end called FRONT.

- Queue can be referred as to be first In first out list.



Complexity of queue :-

	Average				Space comp.
	Access	search	Deletion	Insertion	Worst
Queue	O(n)	O(n)	O(1)	O(1)	O(n)

	Worst			
	Access	search	Insertion	Deletion
Queue	O(n)	O(n)	O(1)	O(1)

Operations on queue :-

1). Enqueue :- Enqueue is used to insert element at rear end of the queue. It returns void.

2). Dequeue :- dequeue operations performs the deletion from front end of queue. The deque operation can also be designed to void.

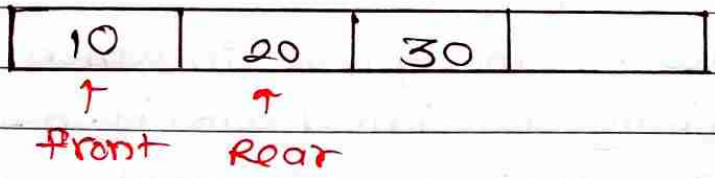
3). peek :- This returns, element which is pointed by front pointer in the queue but does not delete it.

4). queue overflow (is full) :- when queue is completely full, then it shows overflow condition.

5). queue underflow (isempty) :- when there is no element in the queue then it throws underflow condition.

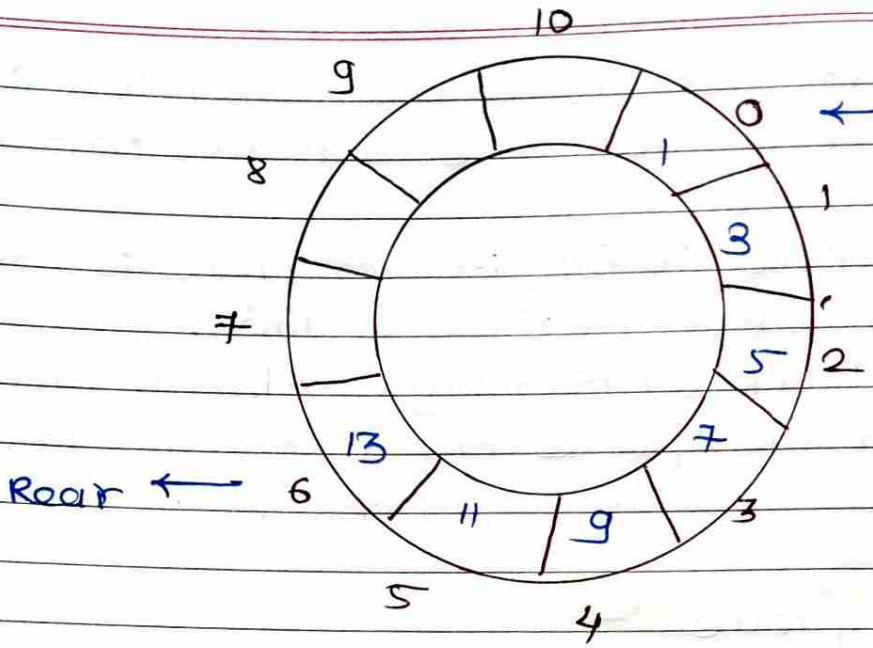
Types of queue :-

1). Linear queue :- In linear queue, an insertion takes place from one end while deletion occurs from another end. It strictly follows FIFO rule. The linear queue can be represented, as shown :



The elements are inserted from rear end, and if we insert more elements in queue, then rear value gets incremented on every insertion.
drawback is using linear queue is : insertion is done only from rear end. The linear queue shows the overflow condition as rear is pointing to last element of the queue.

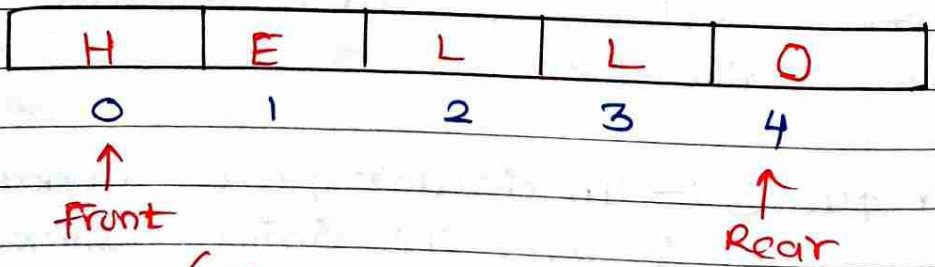
2). circular queue :- In circular queue, all nodes are represented as circular. It is similar to linear queue except that last element of the queue is connected to the first element. It is also known as ring buffer. as all ends are connected to another end.



Drawback of linear queue is overcomes in circular queue. If empty space is available, new element can be added in empty space by simply incrementing value of rear.

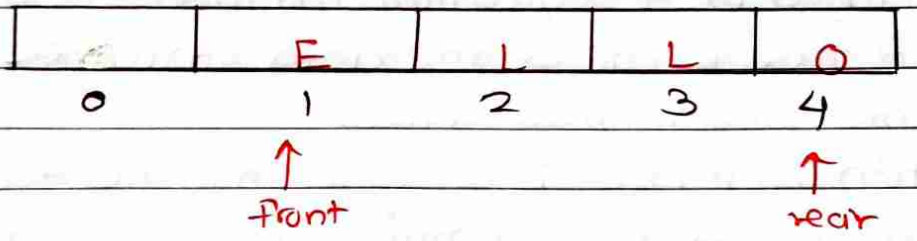
3). Priority queue :— The queue in which each element has some priority associated with it. Based on priority of the element, elements are arranged in a priority queue. If elements occur with same priority, then they are served according to FIFO principle.

* Array representation of queue :—



(Fig : queue after inserting an element)

After deleting element, value of front will increase from -1 to 0, the queue will look like :



(Fig : queue after deleting an element)

Algorithm to insert any element in a queue :-
check if queue is already full by comparing rear to max -1.

Algo: step 1 :- IF REAR = MAX -1
write OVERFLOW
Go to step [END OF IF]
step 2 :- IF FRONT = -1 and REAR = -1
SET FRONT = REAR = 0
ELSE
SET REAR = REAR + 1 [END OF IF].
step 3 :- SET QUEUE [REAR] = NUM
step 4 :- EXIT.

Algorithm to delete an element from queue :-

Algo: step 1 :- IF FRONT = -1 or FRONT > REAR
write UNDERFLOW
ELSE
SET VAL = QUEUE [FRONT]
SET FRONT = FRONT + 1
[END OF IF]
step 2 :- EXIT.