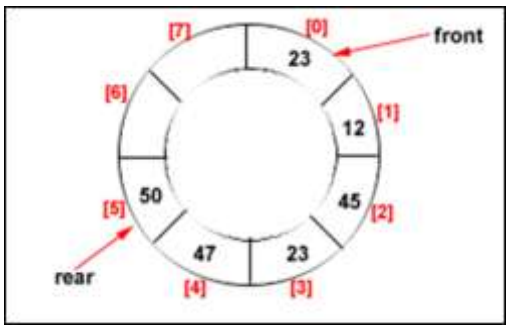
❖ INTRODUCTION TO CQUEUE (FIFO): CIRCULAR QUEUE

Drawback of Linear Queue

Once the queue is full, even though few elements from the front are deleted and some occupied space is relieved, it is not possible to add anymore new elements, as the rear has already reached the Queue's rear most position.

Circular Queue

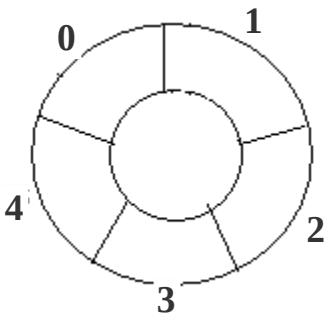
- This queue is not linear but circular.
- In circular queue, once the Queue is full the "**First**" element of the Queue becomes the "**Rear**" most element, if and only if the "**Front**" has moved forward. otherwise it will again be a "Queue overflow" state.



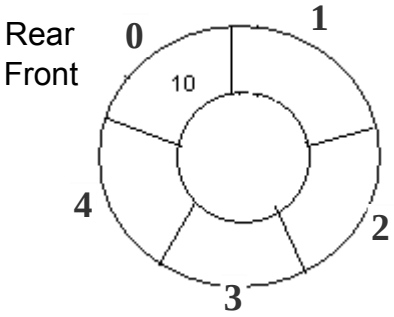
CQ have Rear =5 and Front = 0

Example: Consider the following circular queue with N = 5.

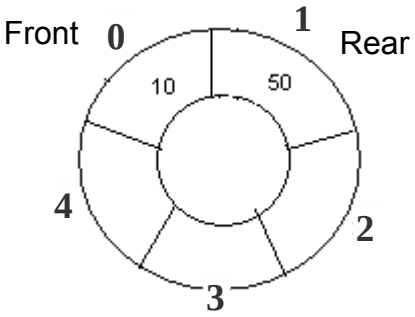
1. Initially, Rear = -1, Front = -1.



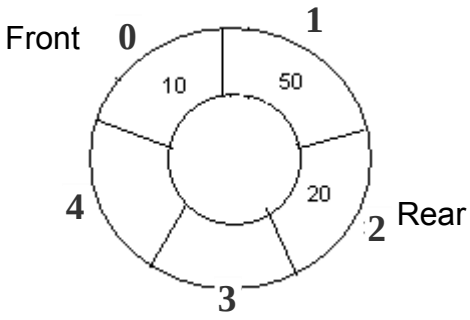
2. Insert 10, Rear = 0, Front = 0.



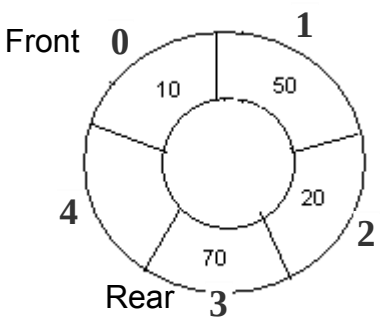
3. Insert 50, Rear = 1, Front = 0



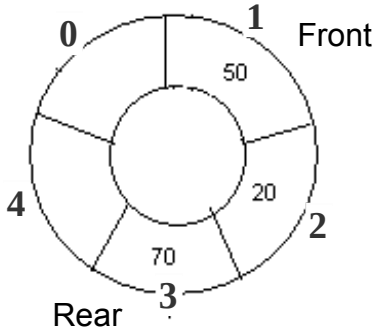
4. Insert 20, Rear = 2, Front = 0.



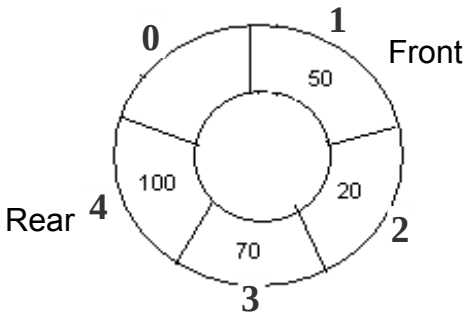
5. Insert 70, Rear = 3, Front = 0.



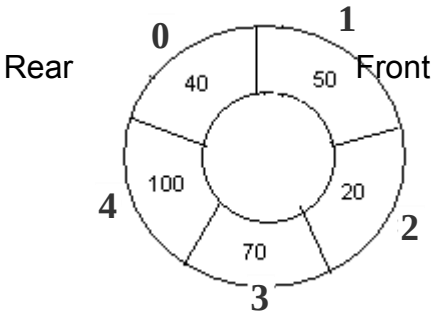
6. Delete front, Rear = 3, Front = 1



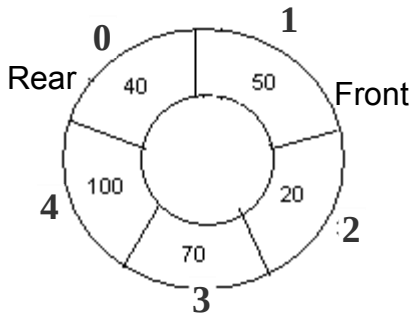
7. Insert 100, Rear =4, Front =1.



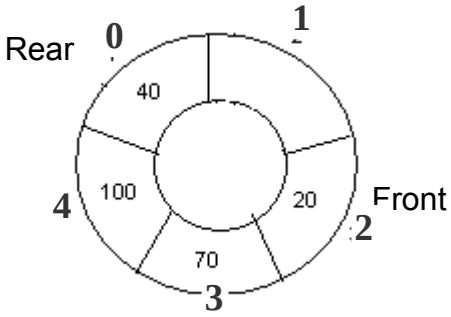
8. Insert 40, Rear = 0, Front = 1.



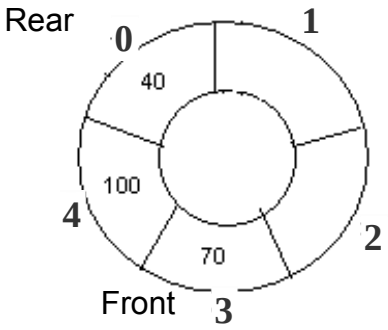
9. Insert 140, Rear =0, Front = 1
As Front = Rear + 1, so Queue overflow.



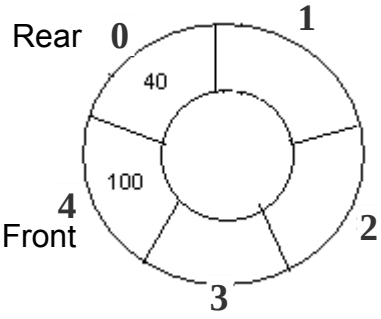
10. Delete front, Rear = 0, Front = 2.



11. Delete front, Rear = 0, Front = 3.



12. Delete front, Rear = 0, Front = 4.



❖ Algorithms for Insert and Delete Operations in Circular Queue For Insert Operation

✓ Insert-Circular-Q(CQueue, Rear, Front, N, Item)

Here, CQueue is a circular queue where to store data. Rear represents the location in which the data element is to be inserted and Front represents the location from which the data element is to be removed. Here N is the maximum size of CQueue and finally, Item is the new item to be added. Initailly Rear =-1 and Front = -1.

Algorithm for Enqueue operation using array

Step 1. Start
Step 2. If (front == (rear+1)%max)
Print error "circular queue overflow "
Step 3. Else
{ rear = (rear+1)%max
Q[rear] = element
If (front == -1) f = 0
}
Step 4. Stop

✓ For Delete Operation Delete-Circular-Q(CQueue, Front, Rear, Item)

Here, CQueue is the place where data are stored. Rear represents the location in which the data element is to be inserted and Front represents the location from which the data element is to be removed.

Algorithm for Dequeue operation using array

Step 1. Start
Step 2. If ((front == rear) && (rear == -1))
Print error "circular queue underflow "
Step 3. else
{ element = Q[front]
If (front == rear) front=rear = -1
else
Front = (front + 1) % max
}
Step 4. Stop