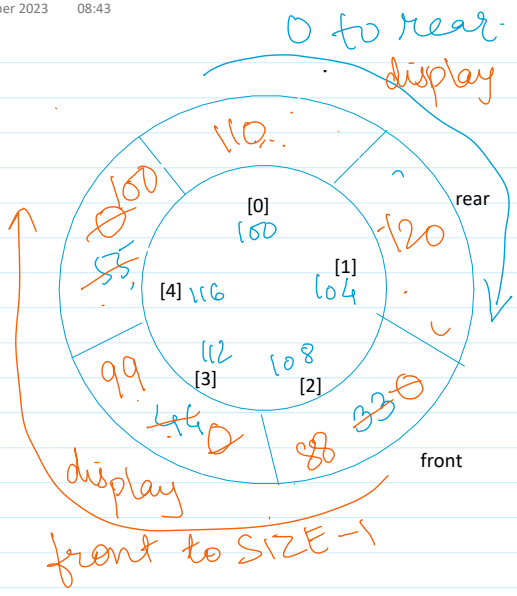
23 October 2023 08:43



rear = -1

rear = (rear+1) %SIZE

(-1+1) % 5

rear = 0 % 5 = 0

(0+1) %5

rear = 1 %5 = 1

= (1+1) %5

rear = 2 %5 = 2

= (2+1) %5

= 3 %5 = 3

= (3+1) %5

4 %5 = 4

= (4+1) %5

= 5 %5 = 0

rear++

rear = rear+1

4+1

rear = 5

rear = (rear+1) % SIZE

Enqueue:

1) Check if queue is not full
2) Increment rear as
   Rear = (rear+1) + SIZE
3) Add element at rear position
4) If front == -1, make front = 0

Dequeue :

Check if Queue is not empty.

Increment front .

If front is 4 and rear is 0

to delete the rear position, we cannot increment front as front ++ Will be index 5 but we want to delete index 0;

So,

Front = front +1 %SIZE

If deleting the last element in queue

If(front == rear)

Front = rear = -1	

Queue empty condition

If(rear == -1) queue is empty

Circular Queue full condition

When Queue is full,

Rear = 0 front 1
Rear = 1, front = 2
Rear = 2 front = 3
Rear = 3 front = 4
rear = 4, front = 0

Front == rear + 1
3 == 2+1

But when rear = 4 front =0
0 == 4+1 does not satisfy the above condition
So the queue full condition can be
Front == (rear+1) %SIZE
0 == (4+1) % 5
0 == 5%5
0==0

To enqueue, instead of rear++
We say,
Rear = (rear+1) % SIZE

Option 2:

```
If rear == SIZE-1
    Rear = 0;
Else
    Rear++;
```