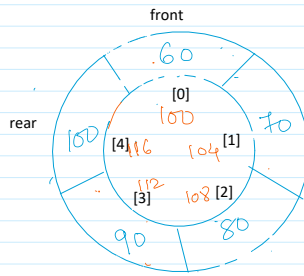


arr[5]

}-1.



| front | Rear |
|-------|------|
| ✓ 1 | 0 |
| ✓ 2 | 1 |
| ✓ 3 | 2 |
| ✓ 4 | 3 |
| 0 | 4 |

Queue Full Condition

$$\text{front} == (\text{Rear} + 1) \% \text{SIZE}$$

$$1 == (0 + 1) \% 5$$

$$1 == 1 \% 5$$

$$2 == (1 + 1) \% 5$$

$$2 == 2 \% 5$$

$$2 == 2$$

$$3 == (2 + 1) \% 5$$

$$3 == 3 \% 5$$

$$3 == 3$$

$$4 == (3 + 1) \% 5$$

$$4 == 4$$

$$0 == (4 + 1) \% 5$$

$$0 == 5 \% 5$$

$$0 == 0$$

Enqueue:

- 1) Increment rear as
Rear = (rear+1) + SIZE
1. Add element at rear position
2. If front == -1, make front = 0

Dequeue :

- 1) 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

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

This means,
Front == rear + 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

Option 2:

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