

rear = (reart) 7. STZE arr Trear (= Jatarear front = (front + 1) 7.5 = (4+1) 7.5 = 57.5 front = 0

front [0] رول[1] [4] (16 77 [08[2]

if (front = = Rear) front = -1 Rear = -1

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

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 = (rearti) % SIZE V Illar = 0 1.5 → 0 4.5=0

(0+1) 7/5->11/5=1 Vear=2 (1+1) 1/5->21/5=2 Vear=2

rear = 2

3+1) 1.5 -> 4.1.5 > 4.

(4+1) %5 → 57.5 → 0 rear = 0

Circular_Queue Page 1

