Queues→

1. A queue is a data structure that follows the principle of FIFO(First In First Out).
2. There are two ends in a queue →FRONT and REAR.
3. An element is always removed from the Front.
4. A new element is always inserted from rear.

Algorithm for inserting an element in a queue→

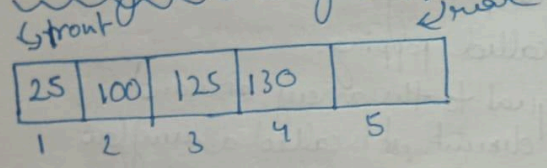
```
1. If(FRONT=1 and REAR=N) or (FRONT=REAR+1),then
   Write : Queue is overflow error
   Return
   [end of if]
2. If(FRONT=NULL and REAR=NULL),then
   Set FRONT:=REAR:=1
   IF REAR=N,then
   Set REAR:=1
   Else
   Set REAR:=REAR+1
[END OF IF]
[END OF IF]
```

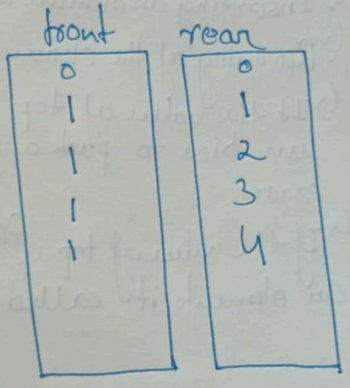
Algorithm for deleting an element from a queue→

```
1. If FRONT=Null,then
2. Write:Queue underflow error
3. Return
4.[End Of IF]
5.Set Val:= QUEUE[FRONT]
6. If FRONT=REAR,then
7. Set FRONT:=REAR:=0[NOW QUEUE IS EMPTY]
8.ELSE
9. IF FRONT=N then
10. Set FRONT:=1
11. Else
12. Set Front:=Front+1
13.[End of If]
14.[End of If]
15.Return
```

↳ A new element is always inserted from rear.

inserting and deleting in a queue

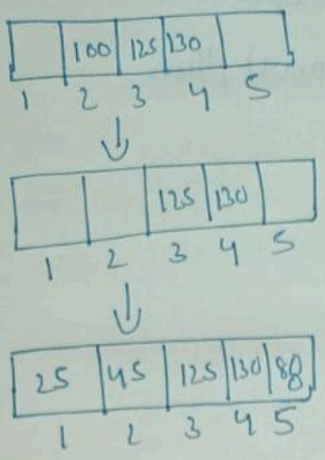




front & rear
↳ 0 ↲
↓
means our queue is empty

↳ for inserting first element set both front, rear = 1

now deleting



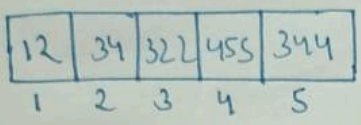
front	rear
0	0
1	1
1	2
1	3
1	4
2	4
3	4
3	5
3	1
3	2

imp

front = rear + 1

↳ indication is queue is full

eg



front	rear
0	0
1	1
1	2
1	3
1	4
1	5
2	5
3	5
4	5
4	4
5	5
1	5
2	5
3	5
4	5
5	5
0	0

34 322 455 344

322 455 344

455 344

344

12 34 322 455 344

34 322 455 344

322 455 344

455 344

344

also the conditional full is front=1, rear=5

size of array

to delete last element make front, rear = 0

→ queue is empty