* Queue :

Queue is a ~~non~~-linear Data Structure in which we perform the Insertion and Deletion Operation.

Queue is follow FIFO structure.
It means First In First Out.

In Queue, whose data enter First this remove by First.

-> Representation of Queue.

Queue represent using the array method, In this method using two pointer variable.

1 Front - Contains the location of First element in Queue.

2 Rear - Contains the location of last element in Queue.

- Queue Operation :

Queue is follow this two operation.

1) Insertion Operation
2) Deletion Operation.

1 Insertion Operation: When an element insert in the Queue, then we use insertion operation.

To Perform the Insertion Operation we have to check "Queue Overflow" condition.

- Queue Overflow: For Queue overflow, Front = 0 and R = max - 1 or F = R + 1.

Before the insert the value in Queue, we have to check this condition.

If Queue is Overflow, then we can not insert element in Queue.

- Algorithm:

1 Start
2 If F = 0 and R = max - 1
then print: Overflow.
return.
3 If F = -1
then set F = 0 and R = 0
else if R = max - 1
then set R = 0

else
set R = R + 1
4 set Q[R] = data
5 End

2 Deletion Operation : When an element delete in the Queue, then we use Deletion Operation.

To Perform the Deletion Operation, we have to check "Queue Underflow" condition.

- Queue Underflow : For Queue Underflow Front = -1.

Before the delete the value in Queue, we have to check Queue is empty or not.

If Queue is empty, then we can not delete element in Queue.

- Algorithm :

1 Start
2 If F = -1
then print : Underflow
return

3 data = Q[F]
4 If F = R
then set F = R = -1
else if F = max - 1
then set F = 0
else
set F = F + 1
5 End.

=> Circular Queue:

In array implementation of Queue, if the rear reaches the last position of Queue then might be spaces are empty in Queue, which can not use in array implementation of Queue.

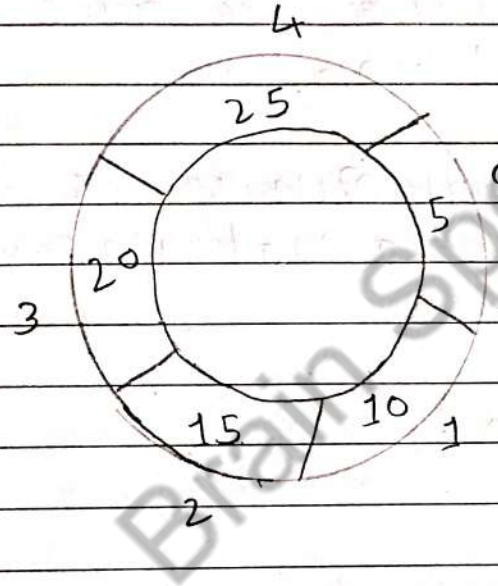
This is one limitation of array represation of Queue.

In Circular Queue, If rear reaches the last position of Queue then might be spaces are empty in Queue, then we can insert the element in Queue.

In Circular Queue, last position of Queue is connected to the first position of Queue.

A Circular Queue is the Form of a circle and it is also know as a Ring Buffer.

Ex.



Circular Queue is also follow this two operation.

1) Insertion Operation
2) Deletion Operation.

1 Insertion Operation : When an element insert in the circular Queue, then we use insertion Operation.

To Perform the Insertion Operation, we have to check "Circular Queue overFlow" condition.

- Circular Queue OverFlow: For Circular Queue OverFlow, F = R

Before the insert value in the circular queue, we have to check this condition.

If Circular Queue is Overflow, then we can not insert value in Queue.

- Algorithm:

1 Start
2 If R = Size - 1
  then R = 0
  else
    R = R + 1
3 If F = R
  then write : OverFlow
4 CQ[R] = X
5 If F = -1
  then F = 0
6 End.

2 Deletion Operation: When an element delete in the circular Queue, then we have to use this operation.

To Perform Delection Operation, we have to check "Circular Queue Underflow" condition.

- Circular Queue Underflow: For Circular Queue Underflow, F = -1

Before the delete value in the circular Queue, we have to check this condition.

If Circular Queue is Underflow, then we can not delete value in the Queue.

- Algorithm:

1 Start
2 If F = -1
then write: Queue is Underflow
return.
3 If F = R
then ~~F = R = -1~~ F = R = -1
4 If F = Size - 1
then F = 0

else F = F+1

5 End.

=> Priority Queue:

Priority Queue is a collection of an element which processed according to their Priorities.

In Priority Queue, the element with the highest Priority would come first in the Priority Queue.

There are two types of Priority use in Priority Queue

1) Ascending
2) Descending.

1 Ascending order Priority Queue:

In Ascending Order Priority Queue, A lower Priority number is given to higher Priority in Queue.

In this Queue, lower number is insert in the Queue.

higher Priority > Lower Priority

lower number > higher number.

Ex.

2	4	6	8

2 Descending Order Priority Queue:

In Descending Order Priority Queue, A higher Priority number is given to higher Priority in Queue.

In this Queue higher number is insert in the Queue.

higher Priority > Lower Priority

higher number > lower number

Ex.

8	6	4	2

Page No.
Date : / /

=> Double Ended Queue:

Double Ended Queue is also called Deque.

In Double Ended Queue, Insertion and Deletion Operation are perform from both ends.

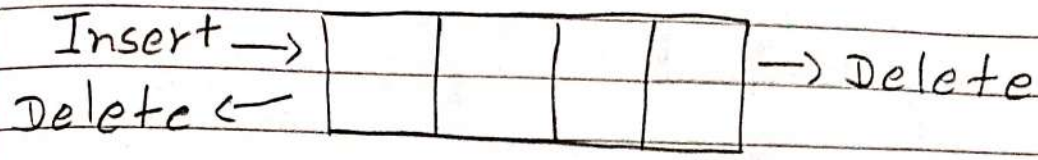
Double Ended Queue does not follow the FIFO rule.

There are two types of Double Ended Queue.

1) Input Restricted Queue
2) Output Restricted Queue

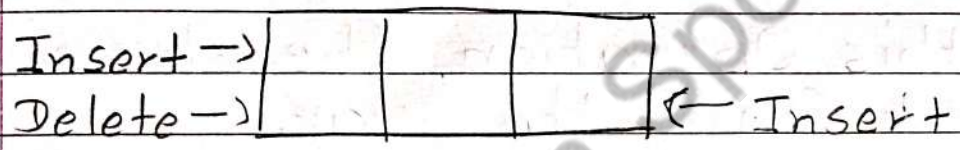
1 Input Restricted Queue:

In this Queue, Insertion operation can be performed at only one end and Deletion can be Perform both ends.



2 Output Restricted Queue:

In this Queue, Deletion Operation can be performed only one end and Insertion Operation can be Perform both ends.



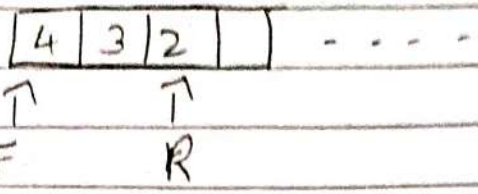
Double Ended Queue is follow this Four Operation.

1) Insertion at Front
2) Insertion at Rear
~~3) Insertion at~~
3) Deletion at Front
4) Deletion at Rear

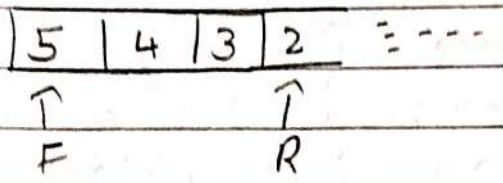
1 Insertion at Front:

In this Operation, the element is insert from the Front end.

Ex.



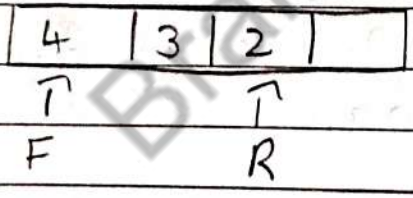
Insert - 5



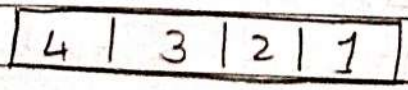
2 Insertion at Rear:

In this Operation, the element Insert from the rear.

Ex.



Insert - 1

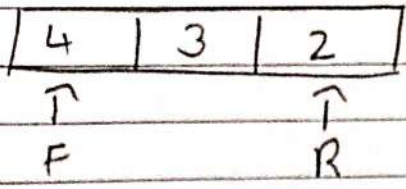


3 Deletion at Front:

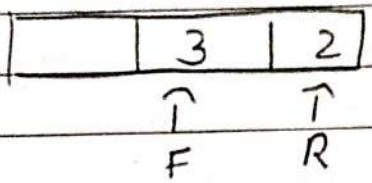
In this Operation, the element Delete from the Front.

Page No.
Date : / /

Ex.



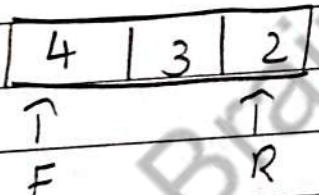
Delete - 4



4 Deletion at Rear:

In this Operation, the element Delete from the Rear.

Ex.



Delete - 2