2.3 The Queue ADT

2.3.1 Queue Model

A Queue is a linear data structure which follows First In First Out (FIFO) principle, in which insertion is performed at rear end and deletion is performed at front end.

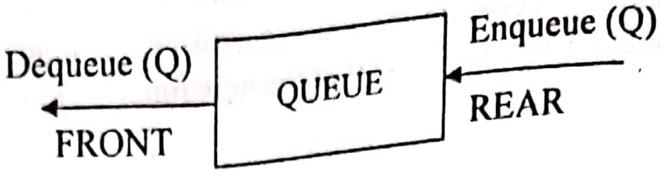


Fig. 2.3.1 Queue Model

Example : Waiting Line in Reservation Counter,

2.3.2 Operations on Queue

The fundamental operations performed on queue are

1. Enqueue
2. Dequeue

Enqueue :

The process of inserting an element in the queue.

Dequeue :

The process of deleting an element from the queue.

Exception Conditions

Overflow : Attempt to insert an element, when the queue is full is said to be overflow cond

Underflow : Attempt to delete an element from the queue, when the queue is empty is said underflow.

2.3.3 Implementation of Queue

Queue can be implemented using arrays and pointers.

Array Implementation

In this implementation queue Q is associated with two pointers namely rear pointer and pointer.

To insert an element X onto the Queue Q, the rear pointer is incremented by 1 and then set

Queue [Rear] = X

To delete an element, the Queue [Front] is returned and the Front Pointer is incremented by

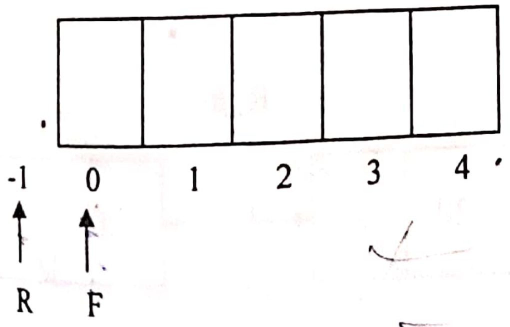
ROUTINE TO ENQUEUE

```
void Enqueue (int X)
{
    if (rear > = max _ Arraysize)
            print (" Queue overflow");
    else
```

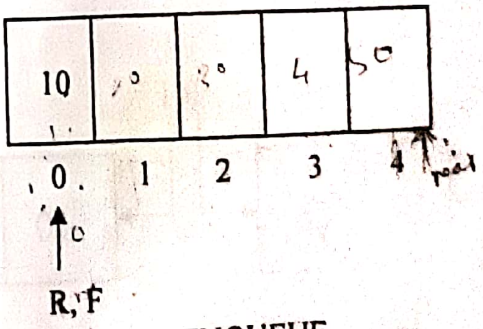
```
        {
                Rear = Rear + 1;
                Queue [Rear] = X;
        }
    }
```

ROUTINE FOR DEQUEUE

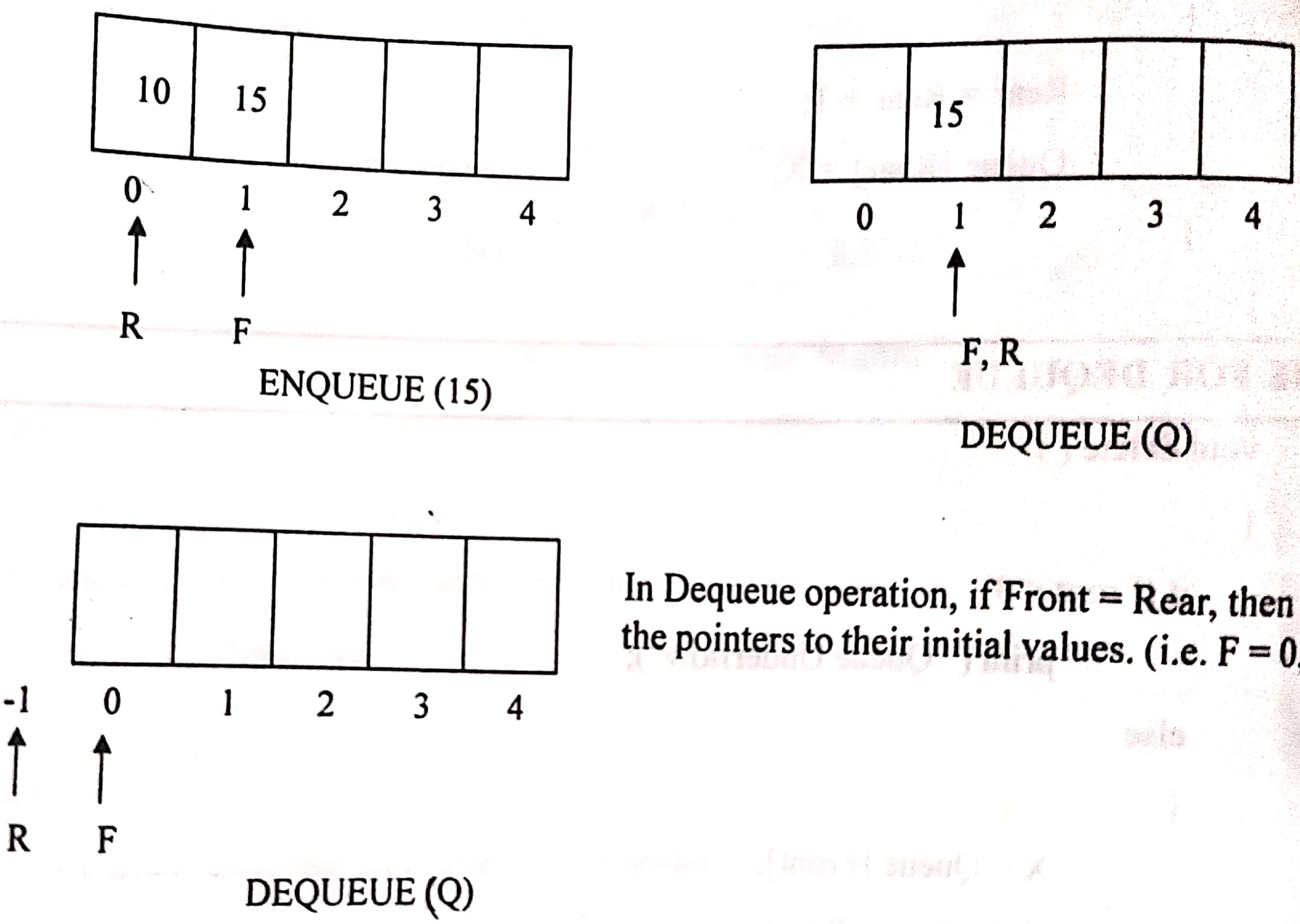
```
    void delete ( )
    {
        if (Front < 0)
                print (" Queue Underflow");
        else
        {
                X = Queue [Front];
                if (Front = = Rear)
                {
                        Front = 0;
                        Rear = -1;
                }
                else
                        Front = Front + 1 ;
        }
    }
```



EMPTY QUEUE



ENQUEUE



ENQUEUE (15)

DEQUEUE (Q)

DEQUEUE (Q)

In Dequeue operation, if Front = Rear, then reset th the pointers to their initial values. (i.e. F = 0, R = -

Fig. 2.3.3 (a) Illustration for Array Implementation of Queue.