

Linear queue.

Algorithm for insert an element in a queue.

```

step1: if Rear = MAX-1
        write overflow
    endif
step2: if Front == -1 & Rear == -1
        set front = 0
        set rear = 0
    else
        set Rear = Rear + 1
    end else
step3: set queue[Rear] = element
step4: exit
    
```

• for delete an element

```

step1: if Front == -1 OR Front > Rear
        write underflow
    
```

```

    else
        set val = queue[Front]
        set front = front + 1
    end if else
    
```

```

step2: exit
    
```

Display:-

```

    if front == -1
        write queue is empty.
    else
    
```

```

        for(int i = front; i <= rear; i++)
        { printf("%d", q[i]);
          }
    
```

```

    end else.
    
```

Circular queue.

Algorithm to insert element

enqueue: To enqueue / insert the element

```
if ref front == rear + 1 or front == 0 and  
    rear == size - 1  
    print queue is full
```

else

```
    if front == -1  
        set front = 0  
        rear = (rear + 1) % size  
        insert the element  
    end if.
```

Dequeue: To dequeue or delete the element from the queue.

```
if front == -1  
    write queue is empty
```

else

```
    val = q[front];
```

```
    if (front == rear)  
        set front = -1  
        set rear = -1
```

end if

else

```
    front = (front + 1) % size.
```

Display:

if front == -1
write the queue is empty.

```
else for (i = front; i != rear; i = (i + 1) % size)
    {
        print("%d", q[i]);
    }
```

o/p for linear queue
enter your choice.

1. enqueue

2. Dequeue

3. Peek

4. display

5. exit

1
enter number:- 3

The inserted no. is 3

enter your choice

1. enqueue

2. Dequeue

3. Peek

4. display

5. exit

1
enter number:- 4

The inserted no. is 4