QUEUE

Queue is a linear Data Structure in which the operations are performed based on FIFO (First In First Out) principle.

In a Queue always the Insertion operation is done at “rear” and Deletion operation is done at “front”.

In a Queue sequence of elements entered into the queue is same as the sequence of elements leave the queue.

Definition

- Queue is a linear Data Structure in which the operations are performed based on FIFO (First In First Out) principle.

Fields

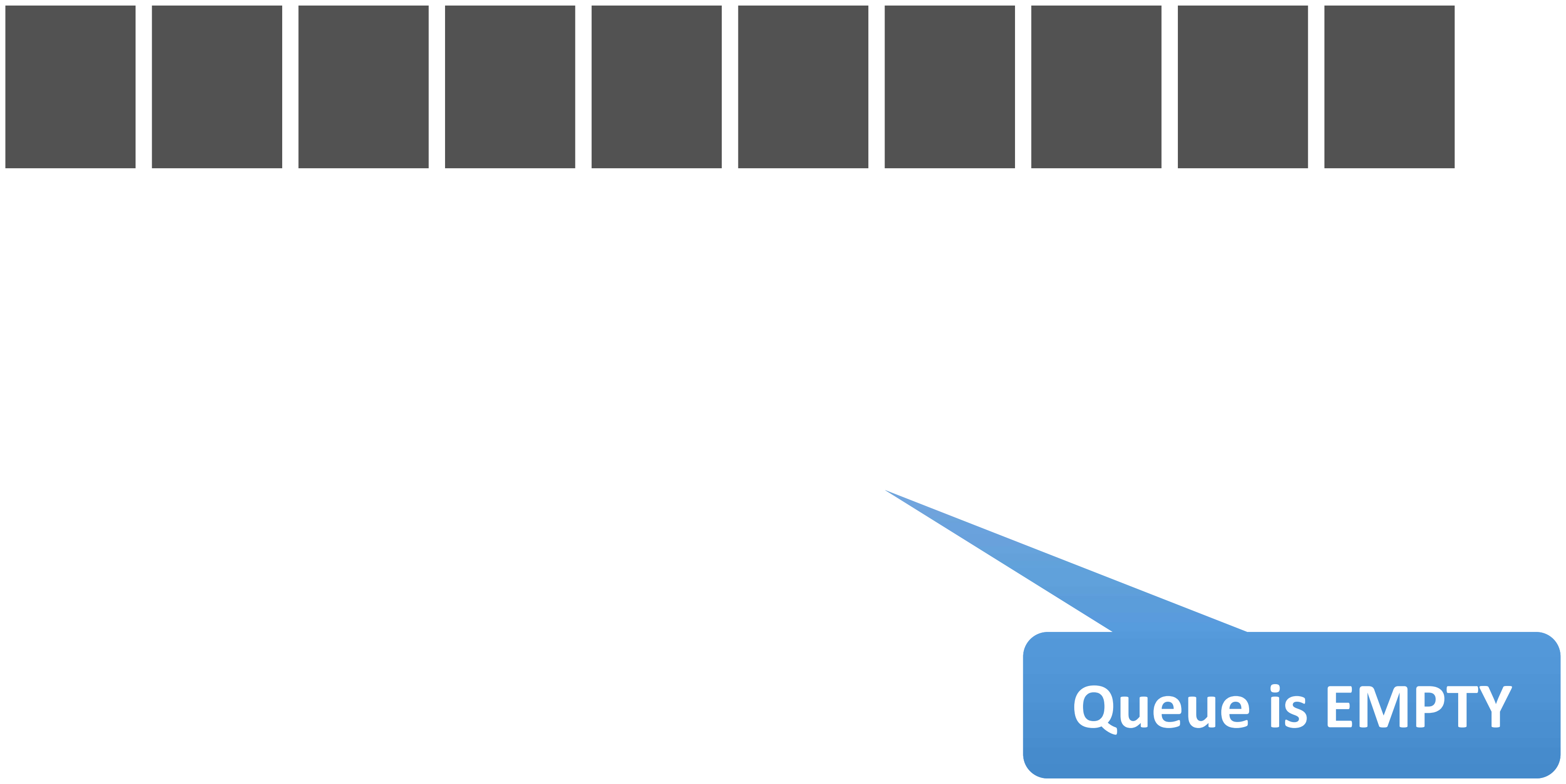
- rear – used to store the position of insertion
- front – used to store the position of deletion
- size – used to store the size of the queue
- element – used to store the value to be inserted

Functions

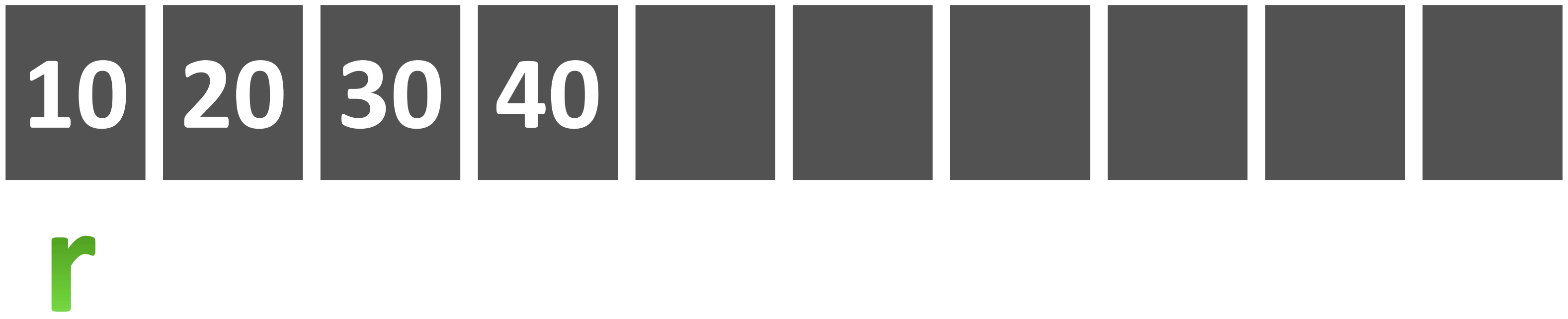
- enQueue(element) – to insert into queue
- deQueue( ) – to delete from queue
- display( ) – to display all elements in queue

Condition

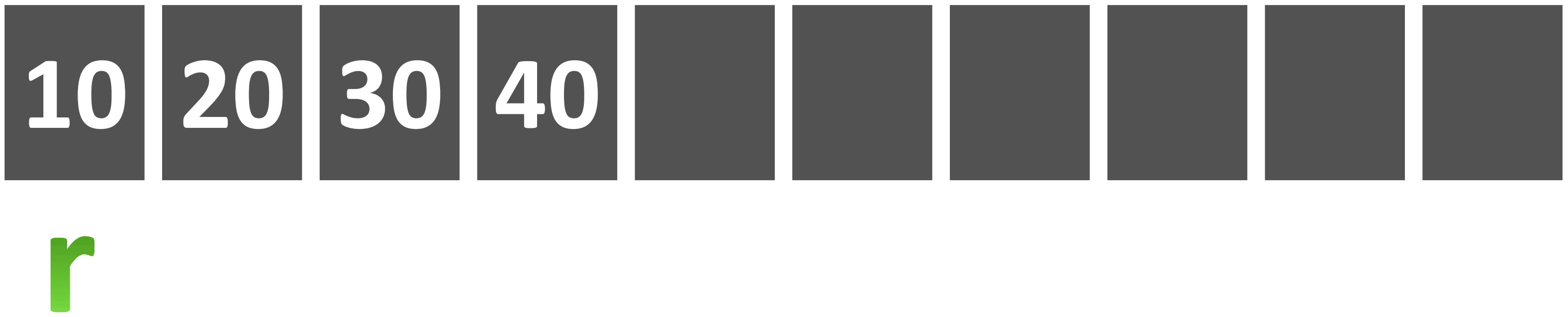
- Rear >= size – Queue is full
- Front = rear = -1 Queue is empty


Queue is EMPTY

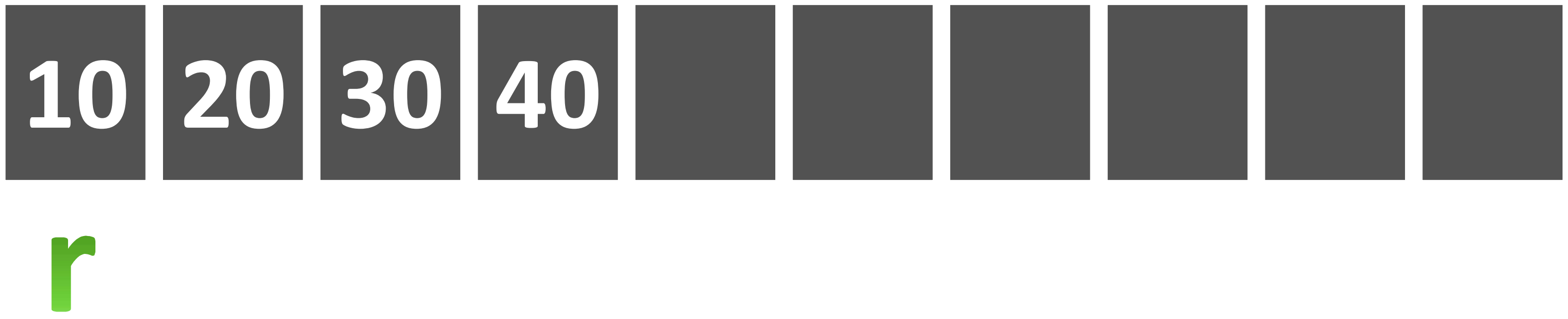
```
void insert(){
int item;
if(rear == MAX-1)
printf("OVERFLOW!! Queue is full \n");
else  if(front ==-1)
front = 0;
scanf("%d", &item);
rear = rear+1;
queue[rear] = item  }}
```



```
void delete(){
if(front == -1 || front>rear){
printf("UNDERFLOW!! Queue empty \n");
return;}
else {
printf("Element deleted from the queue is : %d", queue[front]);
front = front + 1;}}
```



```
void display (){
int i;
if (front == -1)
printf("Queue is empty \n");
else  {
for(i = front; i<=rear; i++)
   printf("%d", queue[i]);
printf("\n"); }}
```



Circular Queue

- A circular queue is one in which the insertion of a new element is done at the very first location of the queue if the last location of the queue is full.

Circular Queue

- We can say that a circular queue is one in which the first element comes just after the last element.
- It can be viewed as a mesh or loop of wire, in which the two ends of the wire are connected together.
- A circular queue overcomes the problem of unutilized space in linear queues implemented as arrays.

Circular Queue

- Bellow show a figure a empty circular queue Q[5] which can accommodate five elements.

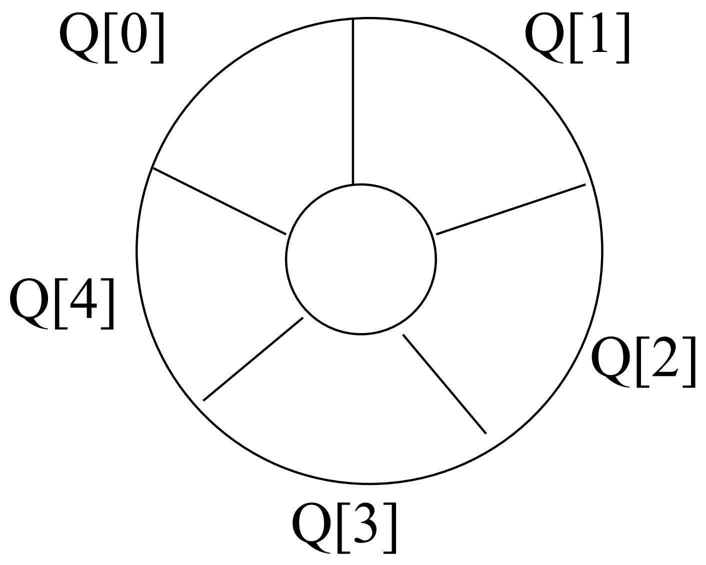


Fig: Circular Queue

Double Ended Queue (Deque)

- It is also a homogeneous list of elements in which insertion and deletion operations are performed from both the ends.
- That is, we can insert elements from the rear end or from the front ends.
- Hence it is called double-ended queue. It is commonly referred as a **Deque.**
- There are two types of Deque. These two types are due to the restrictions put to perform either the insertions or deletions only at one end.

Double Ended Queue (Deque)

- There are:
  - Input-restricted Deque.
  - Output-restricted Deque.
- Bellow show a figure a empty Deque Q[5] which can accommodate five elements.

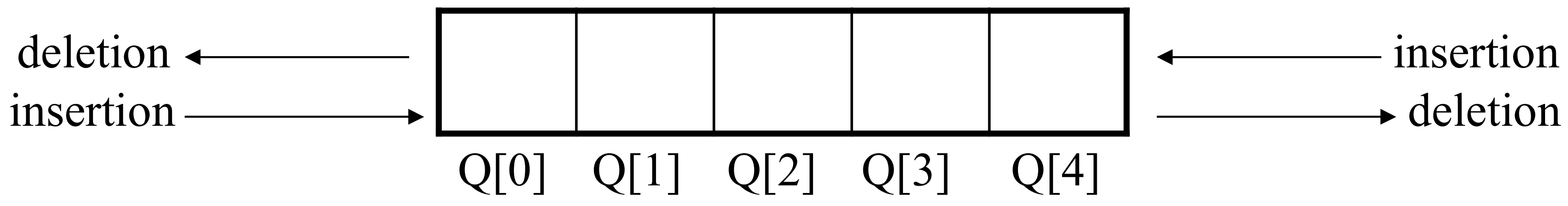


Fig: A Deque

Double Ended Queue (Deque)

- There are:
  - Input-restricted Deque: An input restricted Deque restricts the insertion of the elements at one end only, the deletion of elements can be done at both the end of a queue.

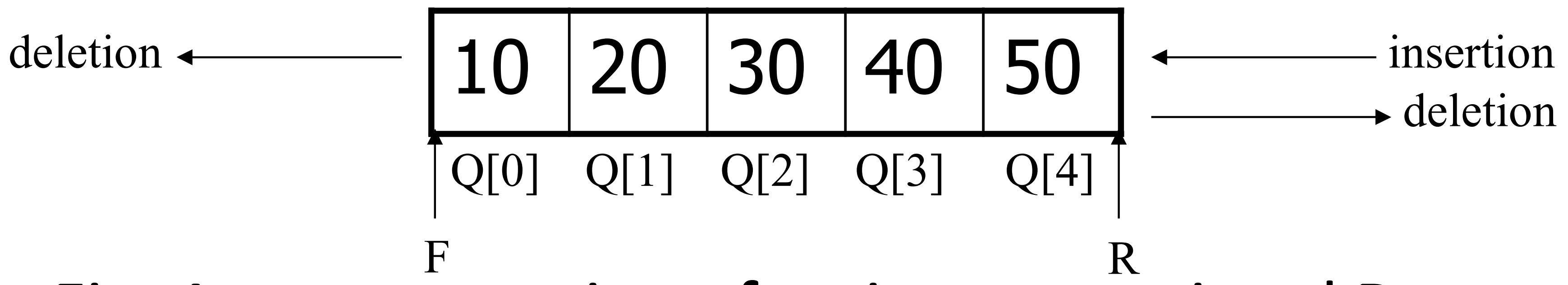


Fig: A representation of an input-restricted Deque

Double Ended Queue (Deque)

- There are:
  - Output-restricted Deque: on the contrary, an Output-restricted Deque, restricts the deletion of elements at one end only, and allows insertion to be done at both the ends of a Deque.

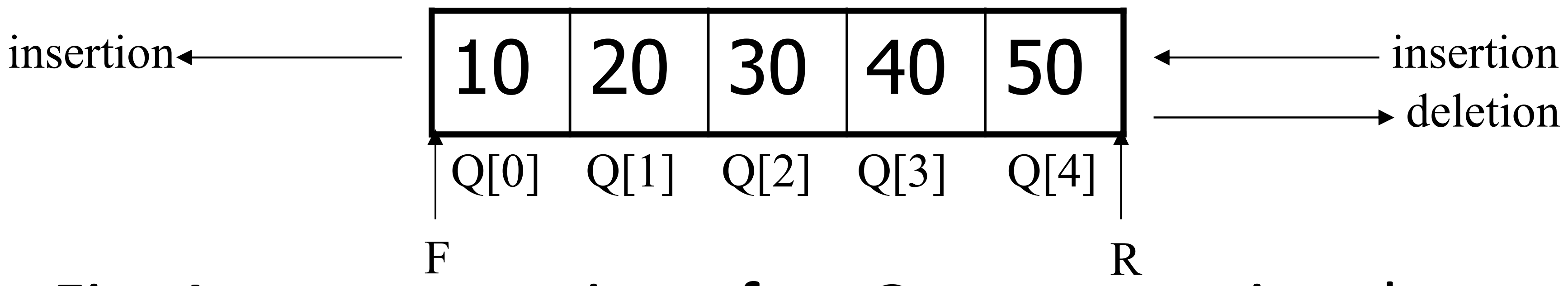


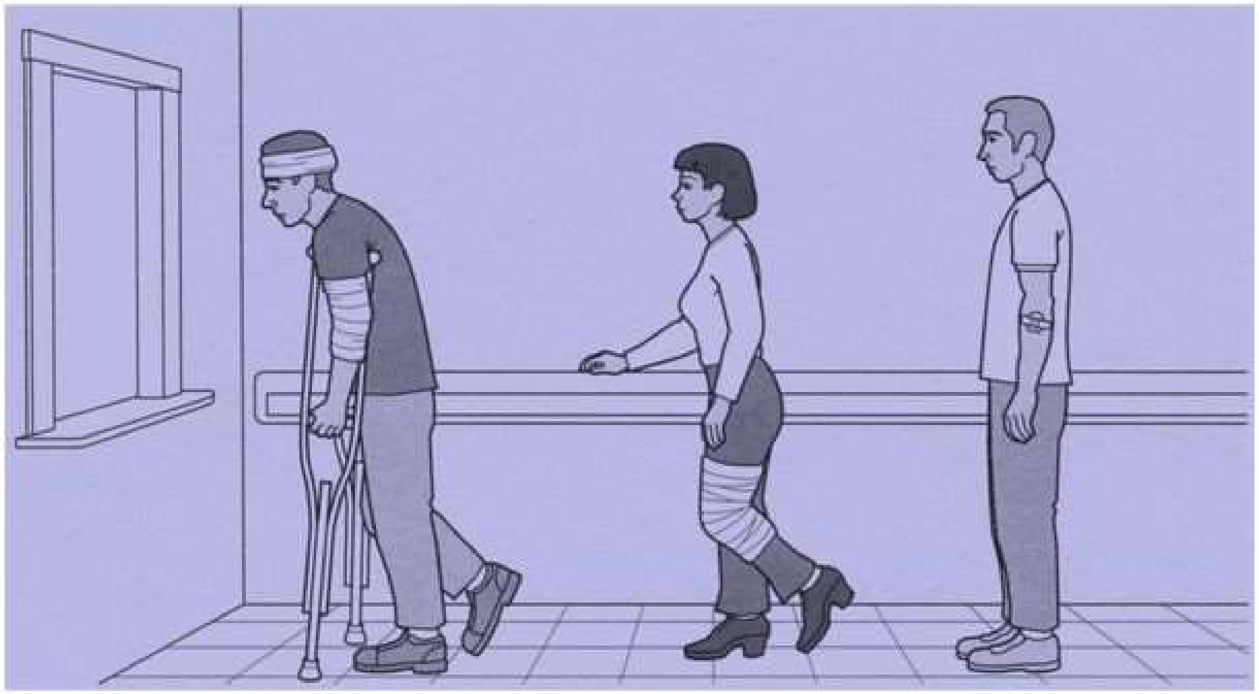
Fig: A representation of an Output-restricted Deque

Priority Queue

- A priority queue is a collection of elements where the elements are stored according to their priority levels.
- The order in which the elements should get added or removed is decided by the priority or the element.
- Following rules are applied to maintain a priority queue.
  - The element with a higher priority is processes before any element of lower priority.
  - If there are elements with same priority, then the element added first in the queue would get processed

PRIORITY QUEUE

- A priority queue is a data structure that supports two basic operations: inserting a new item and removing element with the largest (or smallest) key



Example : queue in hospital
prioritize patient with emergency issue

Priority Queue

- Here, smallest number that is most highest priority and greater number that is less priority.
- Priority queues are used for implementing job scheduling by the operating system.
- Where jobs with higher priorities are to be processed first.
- Another application of priority queue is simulation systems where priority corresponds to event times.