

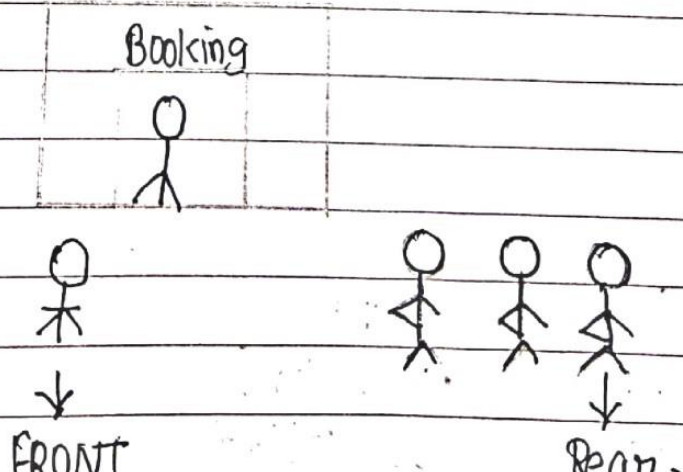
Queues

- Queue is another fundamental data structure that is mostly used in computer science.
- Queue data structure obeys the principle of FIFO (first in, first out) or FCFS (first come first served):

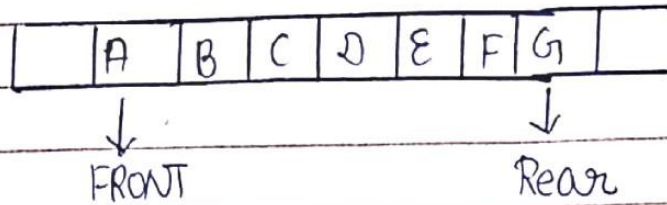
"A Queue is an ordered, homogeneous group of elements in which new elements are added at one end called REAR and elements are removed from the other end called FRONT or HEAD."

Example:-

- Depositing the fee in fee counter.
- Purchasing the ticket in the cinema hall from the booking office.

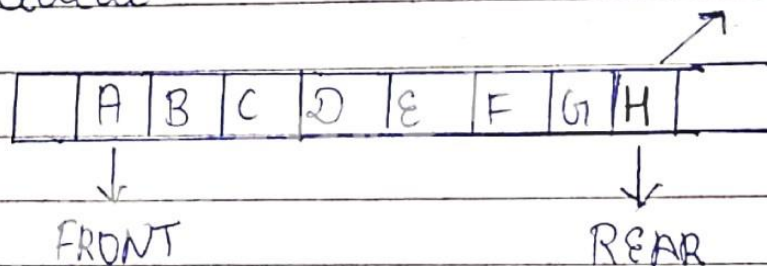


- _/_/_
- Let us consider a queue:-



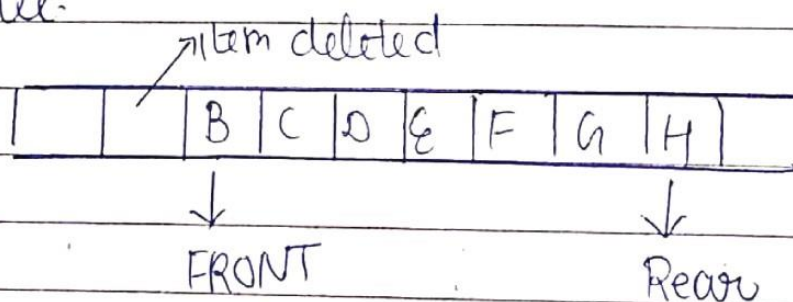
(i) Add item H

- Addition of element is made at rear end of Queue.



(ii) Delete an item:-

- Removal of an item is made at front end of queue.



→ When $FRONT > REAR$, then condition is called Queue Empty.

→ $REAR = N$ (Queue Full).

* Basic Queue Operations:-

There are two basic operations can be performed on the queue.

- 1. Insertion or addition of elements (ENQUEUE).
- 2. Deletion or removal of elements (DEQUEUE).

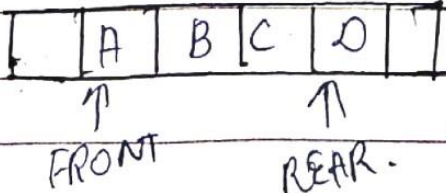
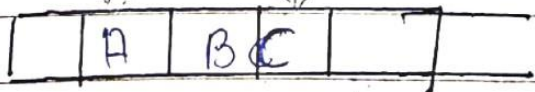
1. Insertion (Enqueue):-

• Insertion of element in the Queue is also known as enqueue operation.

→ New element can be inserted at one end called rear. (when queue is not full).

→ But if queue is full we cannot insert element and 'Overflow' situation occurs.

→ When we insert element the value of REAR is incremented by one.



(After inserting element D)

Algorithm :-

Step 1 :- Start

Step 2 :- If (front = NULL) Then
Write "Underflow" and Exit.

Step 3 :- Set ITEM := Queue[FRONT]

Step 4 :- If (front = rear) then

Set front = NULL

Set rear = NULL

Else

Set front = front + 1

Step 5 :- Exit.

* Representation of Queue in the Memory:-

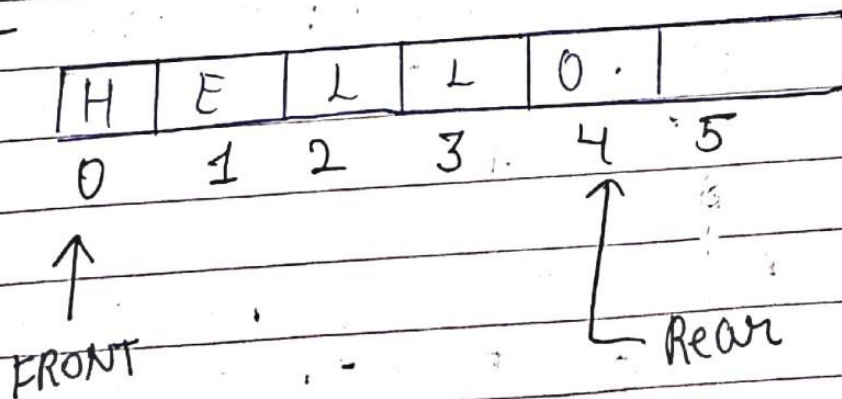
Array can be implemented using two ways:-

1. Queue using Array Representation.
2. Queue using Linked List.

1. Queue using Array:-

- o Array is another data structure that can be used to implement the Queue data structure.
- o There are two variables i.e. front and rear that are implemented in the case of every queue.
- o Initially, the value of front and rear is -1, which represents an empty queue.

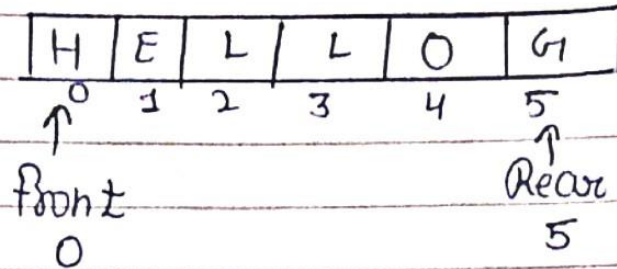
Example:-



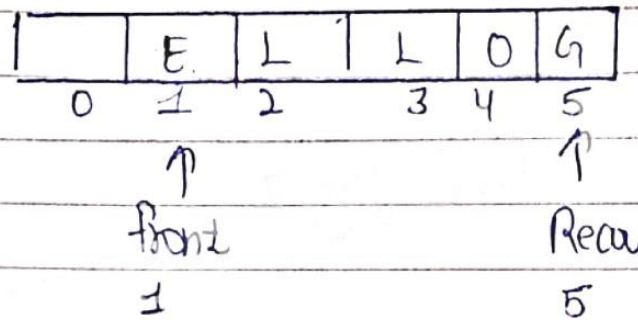
(Queue).

//_

(Queue after inserting an element).



(Queue after deleting an element)



→ If the item is to be inserted as the first element in the list, in this case set the value of front and rear to 0.

Queue insertion Algorithm:

Step 1: if (REAR = MAX) then
write "Overflow and Exit".
[End if]

Step 2: if (FRONT = NULL and REAR = NULL) Then
Set FRONT := 0, REAR := 0

Else

_ / _ / _

Set REAR = REAR + 1
[End if]

Step 3: Queue[REAR] = ITEM

Step 4: Exit.

• Queue deletion Algorithm:-

Step 1: if (FRONT = NULL or FRONT > REAR) then
Write "Underflow" and Exit
[End if]

Step 2:- Set ITEM = Queue[FRONT]

Step 3: Set FRONT = FRONT + 1

Step 4:- Exit.