



Assignment No. 03

①

Title: Circular queue linear data structure

Aim: To implement circular queue using Array as linear data structure.

Problem Statement: Implement circular queue using Array as linear list, perform following operations on it.

- a) Insertion (Enqueue)
- b) Deletion (Dequeue)
- c) Display

Objectives: To understand the simple queue as linear data structure with its limitations  
understand & implement circular queue with array and perform various operations on it.  
Know possible applications of queues.

Outcome: Able to overcome the simple queue limitations by implementing circular queue  
Implement different operations like insert & delete on the circular queue.  
Able to implement real time applications using queues.



## Theory:

### ① Concept of queue as a linear data structure:

Queue: It is an collection of items from which items may be deleted at one end it follows first in first out principle

### ② Simple queue ADT :

enqueue();

using enqueue() method, we can insert element in queue

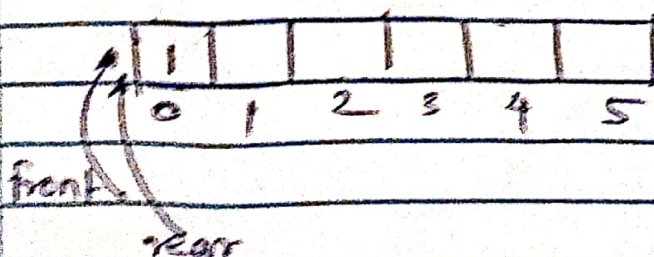
dequeue();

using dequeue() method, we can delete first element which first inserted

isEmpty();

using this method we can determine queue is empty or not

### ③ Graphical representation:





To insert: put new element in location 0 &  
 $\text{rear} = 0, \text{front} = 0$

To delete: Remove from 0th location &  $\text{front} = 1$

Limitation of simple queues & possible solution

In simple queue:

When new item inserted at rear, pointer to rear moves upwards and if item is deleted front moves downward

After a few insert and delete operations the rear might reach the end of the queue and no more items can be inserted although items from front deleted and there is space in queue

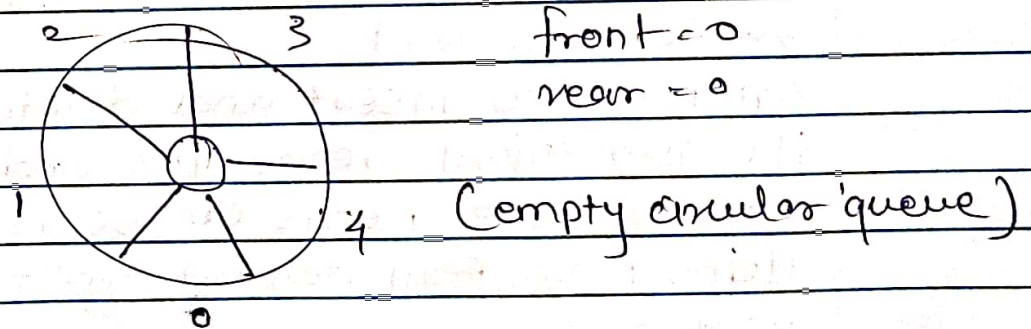
Circular queue and its advantages over simple queue:

Circular queue is linear data structure in which the operations are performed on FIFO principle and last position is connected back to first position to make circle



## Advantages :

In circular queue we utilize memory efficiently. Because in queue, when we delete any element only front is incremented by 1, but that position is not used later, so when we perform more insertion and deletion operations memory increase. But in circular queue, memory is utilized if we delete any element, that position is used later because it is circular.



## Circular queue possible implementation :

To give possible movement inside array, when we go past the last element, it should come back to beginning of the array.

Expression:  $i = (i+1) \% \text{max-size}$

front = (front + 1) % length

rear = (rear + 1) % length



## Application of queue:

Real life examples:

- ① waiting in line  
- waiting on hold for support

Applications in computer science:

- typical use of queues are in simulations for.
- In OS, for controlling access to shared system resource, computer system must often provide a 'holding cue' from message between two process k/a 'buffer' & it's impleted as queue

Validations:

- ① Array size should be in range 1 to max not negative zero or more than max size.
- ② Name must not contain numbers
- ③ Age should be positive

Conclusion:

Analysis of insertion & deletion of operation in circular queue.



Operation		Time complexity
①	Enqueue	$O(1)$
②	Dequeue	$O(1)$