# Queue:

It is highly used data structure where it follows the method of FIFO(First in First out). Queue will be used in booking reservation. Buying products/booking service online.



## How is Queue build

Exit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 3 | 4 |  |

Entry

## Terminology in Queue:

Enqueue- New elements is created

Dequeue- Removal of first.

Isfull- Bollean.

Isempty- Bollean

Delete- Queue

Peek- Last element.

## Types of Queue:

* Linear
* Cyclic

# Linear Queue:

Enqueue- New elements is created

Dequeue- Removal of first.

Isfull- Boolean.

Isempty- Boolean

Delete- Queue

Peek- Last element.

<- Front



<-Rear

Creation of Linear Queue

|  |  |  |
| --- | --- | --- |
|  |  |  |

Enqueue:

Logic for implementing Queue

1. Check if the size is not full
2. Insert the element and increase the size by 1

**Case 1:**

Size =4

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |

Removing last element.

**Case 2:**

Size =3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

After insertion: Size is 4

# Dequeue:

Does the pop functionality in the queues.

Remove the start of the queue.

**Logic**:

Check if the queue is empty.

If not proceed in removal of the start of element and map the next element as start.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 2 | 3 | 4 | 5 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

## Isempty:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Returns Boolean value

## Is Full:

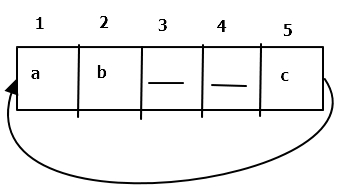
|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |

## Peek:

Returns the top of Queue.

# Circular Queue:

If tail of a queue is tagged to its head, then it is called the circular queue.



**Why Circular Queue:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 7 | 8 | 3(S) | 4 | 5 | 6 |

**Creation and Enqueue of Circular Queue**

Insert an element into the circular queue. In a circular queue, the new element is always inserted at Rear position.

**Steps**:

1. Check whether queue is Full – Check

If it is full then display Queue is full.

1. If queue is not full then,

check if (TOQ == SIZE + 1) if it is true then set rear=0 and insert element.

1. Else,

TOQ++

arr[TOQ]=value

## Dequeue:

Delete an element from the circular queue. In a circular queue, the element is always deleted from front position.

Steps:

1. Check whether queue is Empty.

If it is empty then display Queue is empty.

1. Else if, Check if (start==TOQ) if it is true then set front=rear= -1
2. Else if, Check if (start+1=size), if it is true then set start=0 and return the element.
3. Else, start ++

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 2- S | 3 | 4 | 5 | 6 | 7 - T |

## Isempty:

Check if the any values is available in the circular queue and returns Boolean function

## If empty:

This is a validation function. This is used as an expression to perform different operation in the program

## Delete:

This is the process of deletion of elements from the queue, similar to dequeue. This also involves clearing memory.