Problem statement :-

A double - ended (deque) is a linear list in which additions and deletions may be made at either end.

Objective :-

Understand how to perform insertition and deletion in double ended queue.

Outcome :-

Will be able to perform insertion and deletion in double ended queue.

Theory :-

DeQueue is a data structure in which elements may be added to or deleted from the front or the rear. Like an ordinary queue, a double - ended queue is a data structure it supports the following operations: end_front, enq_front, enq_back deq_front, deq_back and empty.

Deque can be behave like a queue by using only enq_front and deq_front, and behaves like a stack by using only enq_front and deq_rear. The DeQueue is represented as follows.
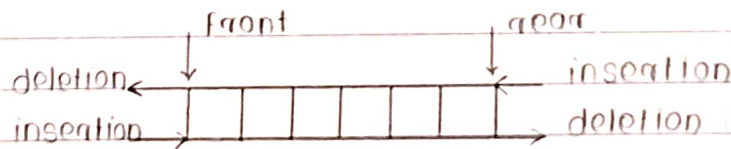
Algorithm :-

Algorithm to add an element into DeQueue: Assumptions : pointer f, r and initial values are -1, -1 Q[] is an array
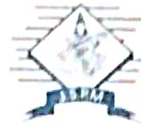
max represent the size of a queue enq_front

Step 1. Start

Step 2. check the queue is full or not.

Step 3. If False update the pointer f as f = f-1

Step 4. Insert the element at pointer f as Q [f] = element

Step 5. Stop enq_back

Step 1: Start

Step 2: check the queue is full or not as if ($r == max -1$) if yes queue is full

Step 3: if False update the pointer r as $r = r+1$

Step 4: insert the element at pointer r as Q [r] = element

Step 5: stop.

Algorithm to delete an element from the DeQueue deq_front

Step 1:- Start

Step 2:- check the queue is empty or not as if ($f == r$) if yes queue is empty.

Step 3:- if False update pointer f as $f = f+1$ and delete element = Q[f]

Step 4:- if ($f == r$) reset pointer f and r as $f = r = -1$

Step 5:- Stop.

deq_back

Step 1: Start

Step 2: check the queue is empty or not as if ($f == r$) if yes queue empty.

Step 3: if False delete element at position r as element = Q[r]

Step 4: update pointer r as $r = r-1$

Step 5: if ($f = r$) reset pointer f and r as $f = r = -1$