

	Problem Statement:
	A double - ended (degue) is a linear list in which additions and
	A double ended (deque) is a linear list in which additions and deletions may be made at either end.
	the transfer of the telephone in the contract of the contract
	Objective:- Understand how to perform insentition and detetion in double
	ended queue.
	Traca que de
2	Outcome :-
	Will be able to perform insertion and deletion in double ended
	queue.
	Theory:
	Theory:- De Queue is a data structure in which elements may be added
	to or deleted from the front or the rear. Like an ordinary
	augus a double - ended augue is a data staucture it supports
TO POR	the following engations: end frant, eng frant, eng back
	the following operations: end_front, enq_front. enq_back deq_front, deq_bock and empty. Deque can be behave tike 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.
A may me	Degue on he behave tike a gueve by using only eng front and
0	doe facet and hoboves like a stack by using only eng-front
	and do again The DeQueue is appaesented as follows.
	and deq. fedt. The beddede to the restriction
	faot about
	(Jeferian)
	insertion deletion deletion
	Algorithm:
	nigorithm to add an element into DeQueue: Assumptions: pointer for and initial values are -1, -1 Q[] is an array
	pointer I, a and initial values age -1, -1 yr 1 is an agray





max represent the size of a queue enq front Step 1. Start slep 2. check the queue is full on not. step 3. If False update the pointer Fas F = F-1 Step 4. Insent the element at pointen fas Q[F] = element step 5. Stop eng_back step 1: stant Step 2: check the queue is full on not as if (n = mox - 1) if yes queue is Full step 3: if False updale the pointer a as a = all slep 4: insert the element at pointer a as Q [7] = element Step 5: Stop. Algorithm to delete an element from the DeQueue deg_front Step 1:- stantale men rei mere tre per mis en e Step 2: - check the queue is empty on not as if (f = = a) if yes queue is emply. slep 3:- if false update pointer fas f = f +1 and delete element = Q[F] slep 4:- if (f == a) neset pointer f and a as f=a = -1 slep 5:- Stop. deg_back step 1: start slep o: check the queue is empty on not as if (F== a) if yes queue empty. slep 3: if False delete element at position a as element = Q[7] step 4: Update pointer a as a = a-1

Step 5: if (f=a) neset pointer f and a ds f=a = -1