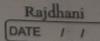
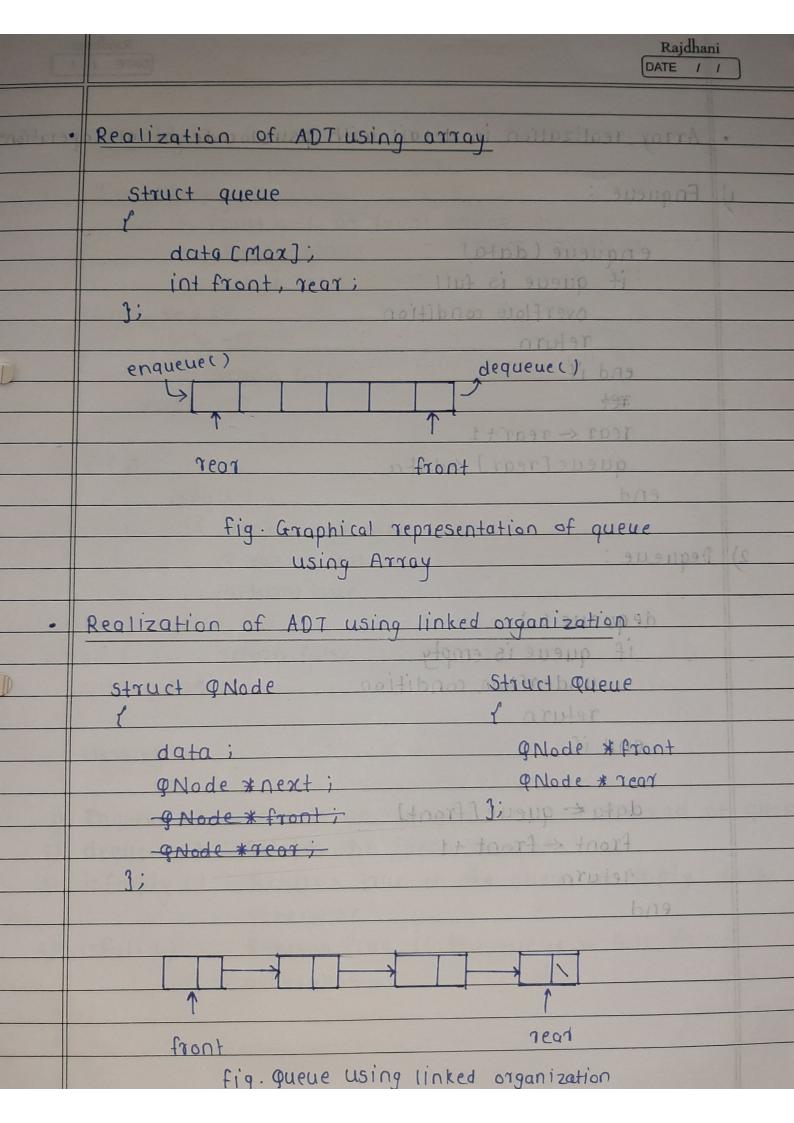
	Title: Assignment 3: Circular Queue Linear data Structure
	Aim: To implement circular queue using Array as a linear
1)	anouadataistructure ne plat rogail à ci ananol-
25	particular order in which operations are performed.
	Problem statement: (0717) 400 taria no teria si reprol
0563	Implement circular queue using array as linear list.
	Perform following operations offit by
	a) Insertion (Enqueue)
	b) Deletion (Dequeue): (vorra 11) TOA susup signis!
	c) Display
	Struct Queue
	Objective:
_	To understand the simple queue as a linear data structure
	with its limitation iron took toi
_	understand & implement circular queue with array & perform
	various operations on it.
-	understand & apply the queue full & queue empty condition
-	know possible applications of queue
	degueue ()
	Outrome:
-	Able to overcome the simple queue limitations by implementing
	circular queue. queue de queue de printerent la contrata de la contrata del contrata de la contrata de la contrata del contrata de la contrata del contrata del contrata de la contrata de la contrata del contrata de
-	Implement different operations like insert & dalala an Iba
	Implement different operations like insert & delete on the
-	
_	Display contents of queue after every operation
	Able to implement real time application using queue
	THOIT FOOT



	Rajdhani DATE / /
	Theory
	J
	Concept of queue as linear data structure:
-	Queue is a linear data structure which follows a
	particular order in which operations are performed. The
	Order is First In First Out (FIFO)
-	It is a special kind of list, where items are inserted
	at one end & deleted from other end.
	a) Insertion (Enquerie)
•	Simple queue ADT (10 Array): (apage 1) noile 100 10
	(c) Display
	Struct Queue
	4 invitation
to pri	To understand the simple queue; [xam] atabdata S
	int front rear; nothing et dear
a sh	- Inderstand of implement circular queue with carrayi & per
	Harrious operations on it
1014	Operations on queue:
	Enqueue () queue de queue
	dequeue ()
	display ()
90019	- Able to everceme the simple queue limitations by imp
•	Graphical Representation of queue
ed!	enqueuec) dequeue ()
	enqueses, adjusted to the policy of the poli
	- Display contents of group of to anery approx
	susup poids neitherity and the translami of old -
	rear front



		DATE / /	DATE / /		
	Argov realization in detail with				
	Array realization in detail with	psuedocade of all open	lations:		
- 1	Enqueue:	Sump tomets			
		(roma alph			
	if queue is full	or thort in			
	Overflow condition				
	return				
	end its anabap	() SNANGNA			
	764	1			
	rear <- rear + 1	1			
	queue [rear] + data	roor			
	end				
	phical representation of queue	fig. Gra			
2)	Dequeue:	211			
	do outdura's 1				
	dequeue (a) ra basail paieur	Realization of AD	*		
	if queue is empty				
	underflow condition	stance andle			
	return	)			
		datai			
		ande seest			
	data = queue[front]				
	front < front +1	- quality corri			
	return				
	euq				
	1				
	reat	tooil			
	in the state agriculture	10.00 0.17			

	CATE / /
3	) is Empty:
	AND AND THE PROPERTY OF MANY AND ADDRESS OF THE PARTY OF
12	(SEmpty ()
	if front = -1 or front rrear
	Empty queue
	Teturn true to the second seco
	end if
	else automorphism and the second seco
1	return folse
	aperquiens are performed based on first to base
- 3	principle & lost position is connected bore to the fire
4)	15 tull
	isfull ()
	if rear = max size
	Queue is full its our susup rations at
o the	Because in queue when we surt muter on elem
593	I too i elseison tout that posies on a from
notton	go Hole Teturn false moraren eu neder oz vetel
1	memory wastone increase bos
3919	- But in circular queue, memory is utilized, if we d
	Lorking sucret that position is used later, because prishable
1)	Enqueue (): Inserts an element at the rear end of queue
2)	dequeue(): Deletes the front element
9	is Empty (): Returns true if the queue is empty, false otherwise
4)	isfull (): Returns true if the queue is full, false
	otherwise.
	enema relucció n'il

· Limitations of simple queue:

In simple queue, if rear is present at maxsize i.e. at last index, though there may empty slots at beginning or middle of queue, queue will be reported as queue full.

The solution is circular queue

circular queue:

Circular queue is a linear data structure in which the aperations are performed based on first In first out (FIFO) principle of last position is connected back to the first position to make a circle.

Advantage of circular queue over simple queue:

In circular queue, we utilize memory efficiently.

Because in queue, when we add delete any element, only front is incremented by 1 but that position is not used

front is incremented by 1 but that position is not used later. So when we perform more add & delete operations

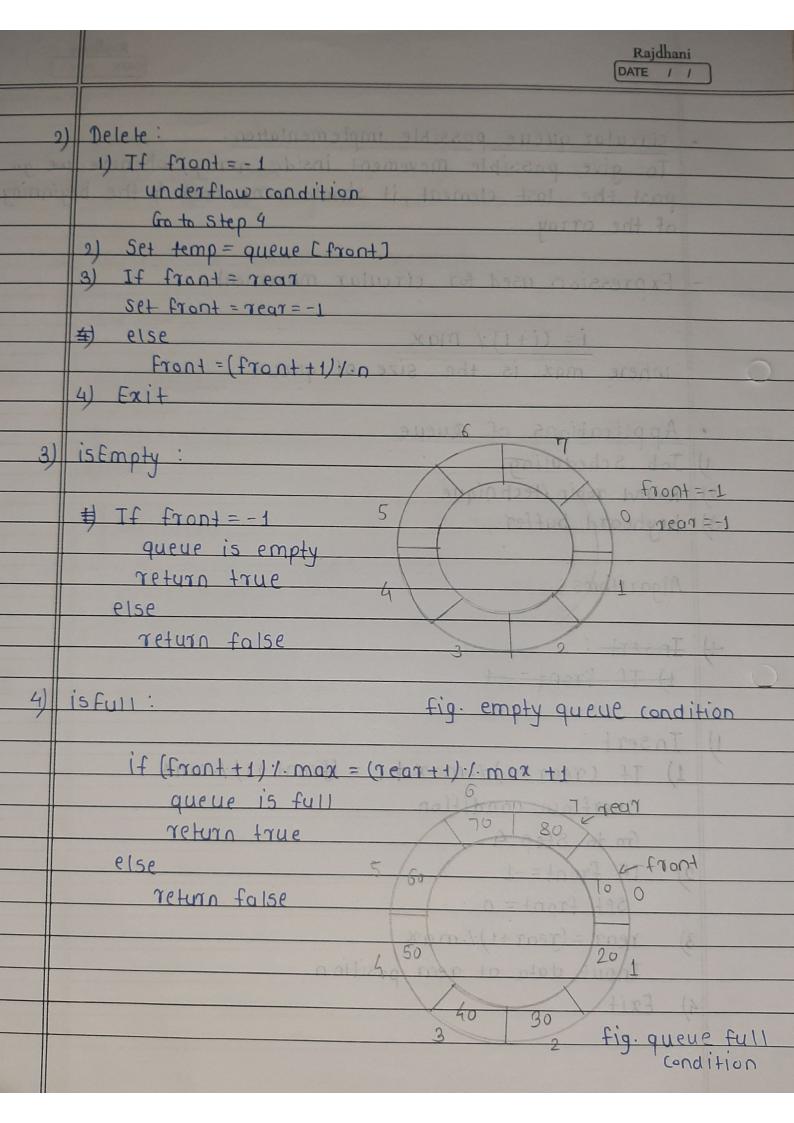
memory wastage increase

But in circular queue, memory is utilized, if we delete any element that position is used later, because it is circular.

Tear - 50

fig. circular queue

		CALL 11	J					
	· circular queue possible implementation:							
	- To give possible movement inside on	riny, wheneve w	e go					
	past the last element, it should come back to the beginning							
	of the array.	22 25 20						
	- Francesia a used for sixuales manne							
	- Expression used for circular moveme	nt sara 11						
0	where max is the size of queue	7= +000						
	Jie or queue.	aliva ta						
	Applications of Queue							
	Job Scheduling	isEmply:	(8.					
	Round robin technique	6						
		= faora at b						
		queue is						
	Algorithms:	t arular						
		9219						
4)	Insert:	arujer -						
	t) If front = -1							
1307	Fig. emply queue condi	isfall:	14					
1)	Insert:							
	1) If (rear +1) / max +1 = (front +1)	1./. max +						
	Overflow condition IIII di							
	Go to Step 4							
	2) If front = -1	9219						
		anular						
		HIDITA						
			1					
	Input data at rear position							
	4) Exit							
1	Fig. queu							



					4		6
1 30	~	8 4	4	24	10	n	
- X	ш	1.2	ш	$u_{\perp}$	LU	201	

1) Array size should be in range 1 to maximum It should not be negative, zero nor more than maximum size.

2) Patient id Should not repeated.

TEBLY ODISES:

Conclusion:

Analysis of insertion & deletion of operations in circular queue:

Operation	0	Time Complexity			
		, ,			
1) en que	ue()	0(1)			
21 Deque		0(1)			
/					