

Scanned with Cam

*	Insertion happens at the great end.
*	Removal happens at the front end. Le front rear
	Removal happens of rear
	Now if we want to do removal, simply
	move front cheed:
	front pear
	5 6 7 11
	And
Note	Queues are majorly used in graphs,
	traversals and sliding window approach.
-	push (9) front rear
	5 6 7 11 9
	Licina C.T.I
	Using STL We need to include header file i.e
	# include <quene></quene>
	The factor of th
(1)	Creation of queue
	queue <int>q;</int>
	The above queue will store integer value
(ii)_	Inserting element in queue
	q. push (5);
	9. push (10);
	q. bush (IS);
(iii)	Remove element in queue
(11)	0.bpb(.)
	Here 5 will be removed from the queue.
	from the queue

Scarineu with Cam

(iv)	Checking empty or not
	9. empty () > True if size = 0 else return
4	False.
(v)	Size of queue
	cout << q. size() j →2 will be the size here
(vi)	Checking front element of the queue
	Checking front element of the queue cout << Q. Front () i → 10 will be the anshere
	Ly to the Palace in the
	Implementation of queue using array
	Implementation of queue using array & We will be requiring front, near, array &
	size.
	Initially front = 0 and Hear = 0
	Alman a manage a fil
(i)	Push
1	Check queue is full or not. If not empty, insert
	if (rear = = size) { // Condition of full
,	cout << "Queue is Full";
1.0	3
	else {
7	arr [rear] = data;
	rear ++j
	3 - man dans dans dans dans dans dans dans da
<u>(ii)</u>	Pop
	First check if queue is empty or not. If not
	empty move front ahead.
	if (front == rear) { cout << "∈mpty";
	cout << "Empty"
	<u>5</u>
	else {
Viol	

N	Page
	arr [front] = -1;
-	
	Check if queue empty or not. If empty then make default behaviour
	Check it queue company be haviour
	empty then make de
1801	if (front = = near)?
	front = 0; ¿ Default behaviour -
	rear = 0) Joutilize the space.
	Just the tree to diffige the space.
• • •	1 set with the surface of the second
(111)	get Front ()
	Check empty queue or not of queue is -
J 15	empty, then simply print empty else
	return and [front].
	the world have all done in white in
	if (front = = near){
	cout << ((Empty")
- 1-44-1271	3 - 3 - a Thetan a stuff and an are
	else {
	return any [front];
	3
<u>(iv)</u>	
	tront = = Henr = Empty
	return follo
	return false.
(V)	get Size ()
	Gimbly return you
	empty location. Pointing to the front
	1 2 3 4 5 6 Juean
	DUGILLEU WILL

Scarified with Cam

	IS
	year-front size. Hence 6-0 = 6 is the
	Size
	1 CF 21 F D To Y
	Circular Queue
	Here let's implement with front = - 1 and
	rear = -1 as initial things.
	industration in the second second
	front near
	We need to handle insertion of 1st element
1 1	explicitly when the queue is empty.
	a travalti in care training the second of th
	Push operation
	Full - Then we can't insert and hence
	simply display message.
(2)	First element insert > Front ++ , year ++&
	then arm [rear] = data;
(3)_	Establish circular nature > when mean == n-1
	22 front 1=0, then make rear =0 and
	then aur [rear] = data
(4)	Normal case > Hear ++ and arr [rear]=data.
	void push (int data) { // Queue is full (One condition pending)
-	if ((front = = 0 & & near = = size-1) 1 (rear = = (front-1)% (size-1)){
	cout << "cannot insert"; 3
	//Single element case
	else if (front = = -1) {
	Front = Mean = 0;
	arr [rear] = data;
	<u>4</u> 12 22 16 15 24 15 11 24 11

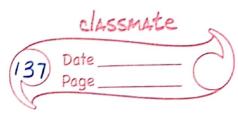
```
else if (front 1=0 && year == size-1).
                 rear = 0)
        normal flow
      else {
          arr [rear] = data;
          9 eas ++ )
               rear tront
                            One more case in
                            which queue is
               4
                   1
            3
                            full.
(IV Pop operation
  Check for empty
   Single element
    Circular nature
(4)
    normal flow
   Void
         bob () {
          / Empty check
            (front = = -1)
               cout << " Empty queue";
        // Single element
                (front = = year)
               ann [front
               front = year =-1;
        /Circular nature
              (front = =size-1){
       4) Here we don't have to check rear =0
                              Scarineu with Cam
```

	front = 0 i
	3- 1- 1- 1- 1- 1- 1- 1- 1- 1- 1- 1- 1- 1-
	else { // Normal flow
	5ront + + ;
W #	was 3 of de ged do don't be all a son
	3
	1/p restricted queue
	insertion from rear end.
	removal from both rear end & front end.
	0/p restricted queue
	insertion from both end i e front & rear.
	removal from front end only
	101-syle= 1010x (= 0-= 10.)
	Doubly ended quine (deque)
	It is pronounced as deck. We can do bush
	and pop both operations from both ends i.e.
	front and rear end.
	LUDSE YOU ILC
(i)	push Rear
	Same code as that of push operation of
	the circular queue Circular condition wile
	depend on the question.
	(U) Creah un
(11)	bush Front
	Full condition & single (first) element (ase
1	will be sa me.
	Circular nature
	front == 0 & & rear 1 = 5131-1, then
	make front = Size -1
-57-	

ocanneu with oam

	17 - datai
	Normal flow arm [front] = data; Instead of rear ++, simply do front.
	Inchead of rear ++ simply as front.
(111)) popfront () that of bob of the circular
	Some code as that of pop of the wich
	quue.
	pop Rear () single
	Empty and element case will be
-	same.
	Circular nature
	rear == 0 => rear = size-1;
	- Compati muno bahas plansi
<u> 1</u>	Normal flow
8 1 1 1	Simply do rear do man de
	STL for deque
	First of all we need to include the
12	header file.
	#include (deque)
	constant and an land
	Creation
52 (ii)	deque <int>dj</int>
2 (11)	push operations in deque d.push_front (5);
	d. push_front (10);
	d. push_back (15);
	d. push_back (20);
	10 5 20 15
	10 5 20 15

Scarineu with Cam



```
(iii) Size of deque
cout << "size = " << d size();
(iv) Checking front & Hear
cout << d.front (); // Front element
    cout << d. back () j // Rear element
    Checking empty or not
dempty () -) True if empty otherwise false.
```

Scanned with Cam