	30/04/2023
QI	Reverse a queue.
y <mark>a</mark>	
<u> </u>	1/p → 3 6 9 2 8
	$0/b \rightarrow 8 2 9 6 3$
i	Approach-1
	Simply create a stack and pop each element
1	of the queue & Bush in stack. Now popelement
	from stack & Bush in the queue
30	Lite of Valldamesia margarita II
1	Time complexity = O(n)
4	Time complexity = O(n) Space complexity = O(n)
	Abbroach-2
	We can reverse the gueue with the help of
	recursion also. We will just solve one case &
110	rest recursion will handle.
	(3) 6 9 2 8
	Recursion
	pop out but save it

Scarned with Cam

1)	Save front element and popit.
<u>a)</u>	Recursive call for rest of elements.
	8 2 9 6 4 4 4
3)	Now simply bush the front element which was saved.
	8 2 9 6 3
	Hence the queue has been neversed.
	Code Code Code Code Code Code Code Code
(i)	Using stack
(12	USING STACK
	void reverse Queue (queue (int> & q) {
	// Create Stack
	stack <int>stj</int>
+0	// rush queue elements to class
el grad	Wille (19.empty ()))
	int element = q. front ();
	DC PUSh (Plemant) · // T
	3 9. pop(); // Remove from queue
	1/Push clan at 1
	// Push elements from stack to queue While (Ist-empty ()) {
	intelement - al
	int element = st·top(); St·pop();
	q-push (clement);
	3
	3
,••	
(i i)	Using recursion

Scarineu with CaM

	void reverse Queue (queue <int) &="" q){<="" th=""></int)>
1	// Base case
	if (a.empty()){
	retwinj
	- 1 3 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -
	//Save front element
	int element = q.front();
	9. pop();
	1/ Recursive call
	reverse Queue (q) i
	// Push back the element
	a.push (element)
	3
Q2	Reverse first k elements of queue.
	1/b-1 3 6 9 12 15, k=3
	0/p - 9 6 3 12 15
	First insert the k elements in the stack.
	5.131×2 13
	9(12 15
	6.
	3 (3/15/11-11-11-11-11-11-11-11-11-11-11-11-11-
3)	Simply popelements from stack & Bushin
	QAIQIAQ .
	$n \rightarrow n$
	12 15 9 6 3
	(n-k) (K)
3)	First pop n-k elements & then pop in the
	queue.

Scarineu with Carn

	9 6 3 12 15
	Code
	(nibug 9)
	void reverse K Queue (quue (int) & q, int k){
	// Creation of stack Stack <int>si</int>
	int count = o;
	int n = q.size())
	"First insert & elements into stack
	while ([q.empty()){
	// Fetch front element
	int temp = q.front();
	2. pop();
	S. push (temp)
	count ++ >
	if (count == k) { // Relements to be
	3 break; pushed
. 9	0+03 in alexander
	// Push elements from stack to queue
	while (Is compty ()){
	Int element = s.tob();
	5.00(),
	q.push (element);
94	1/D - 1 11 (
	// Push the (n-k) elements count = 0;
	while (10^{-k})
	while (1 q. empty ()) { 4 To handle k=six
,	int temp = q. front (); case
	q. push (temp);
	II.

Scarined with cam

		4
		count ++ i a se le surre locie co Il
-17		if (count = = n-k) { // n-k elements to
		breaki be pushed
4.		3 14 08 56 4 314
		3
		Time complexity = O(n)
		Space complexity = 0 (n)
		Extra conditions as a line of the
		if (R<=0 11 R>n) & Do nothing in this:
2		retwin j J case.
	0.	
	W 3	Interleave 1st and 2nd half of the queue.
-		i/b → 10 20 30 40 50 60 70 80
1 1		0/p + 10 50 20 60 30 70 40 80
		Interlegging means 1st element of queue will come
_ فر		and then element after mid will come 4 so on.
,—		
J-		1st half - 10,20,30,40
<i></i>		1st half - 10,20,30,40 2nd half - 50,60,70,80
_		
1-		Seperate out the 2 halfs.
<i>j</i> —		New queue - 10 20 30 40 3/st half of Original anene
5	-	original queue 7 50 60 70,80 Original queue
1.	01	Distribution to account to account to
	-3)	Pop & push element from new queue to Original queue. Then pop & push element from original
		queue. Then pop & push element from agrica
		queue to the original queue
1		

Scarned with Carri

	Classmate
142	Date Page

(i)	original queue → 50 60 70 80 10
Ø**	new queue - 20,30,400
.,	Spirit T
(11)	0Q - 60 70 80 10 50
	NQ - 20 30 40
(111)	0Q 7 60 70 80 10 50 20
	NQ - 30 - 40 May - 1 description
(iv)	0Q + 70 80 1050 2060
	NQ - 30 40
(V)	0Q - 70 80 10 50 20 60 3000
	NQ 7 40 la una har har la complete
(VI)	00 10 50 20 (2 3 7
38	NQ 7 40
(vii)	0078010 5- 2
(VII)	0Q + 80 10 50 20 60 30 70 40
	i, supposed to a south the
Viii	02-10 50 20 60 30 70 40 80
	08 70 40 80
	Code
	The specific and the second
- \	void interleave Quine (queue Kint>40Q) {
Jan 1 Jan 1	J EM MILLS
	if (OQ. empty ()) { return;
TARALLA A	3
	int n = 0Q·size();
	int $k = n/2$
	int count = 0;
	Scarnieu wini ca

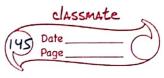
Scarined with Cam

54.75	
1.	queue <int>& nQ;</int>
	// Push half elements of old to na
	while (10Q.empty ()) {
H)	int temp = 0Q.front();
	09.000();
1	na·bush (temp)
, tak	count ++;
	// Done half elements -> break
1	if (count = = k) {
1.4-	break; 8.2
, N	3 (31-131-11-13-13
	3
	// Start interleaving
	while (10Q.empty() && Ina.empty()){
281	// Push from na to 0a
	inf first = nQ.front();
TAN-	na.pop();
	o Q. push (first)
	11 Push from 00 to no
	int second = OQ. front();
	0Q.pop();
	oQ. push (second)
	Jan a signa da san sa gaba ya malanyi
	//Handling odd case
	if (n & 1) {1/0 Q has one extra element than nQ
	int element = oQ.front();
	OQ. popa() igning and again again
in .	0Q. push (element)
	2
	Secretary and the course of an interior
Wa!	and Clubs and the distance A also also
مجما	e - Checking odd or even with the bit wise 4 operator

ocanneu wini cant

	3 → 11
	11 & o1 = 1 & True - Odd
	4-100 - 11 June 10 5 line
	100 & 001 = 0,00 = 0 3 False → Even
_	i () nod-vi:
<u>Q4</u>	First negative integer in every window of
	Size k. (Process first window and then process) other windows via insection & deletion
	Other windows via insertion & deletion
	i/b→ 12 -1 -7 8 -15 30 16 28
	R = 3
	$0/p \rightarrow \{-1,-1,-7,-15,-15,0\}$
	4 no negative
	Sia 1 72 ; integer
	$\{12, -1, -7\} \rightarrow -1$
	$\{-1, -7, 83 \rightarrow -1\}$ $\{-7, 8, -153 \rightarrow -7\}$
	$\{8, -15, 303 \rightarrow -15\}$
	{-15,30,163 → -15
	30.16.283 → 0
	(20) 70 9 4 85 O 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	Approach-1
	9f the input is a vector in
	9f the input is a vector, then we can do the question by using 2 nested loops but the
	time complexity will be O(nxk).
de i	$\partial BCO(nxR)$.
	Abbroach-2
	Here when we are anim
	then front element is getting removed whereas is fossible in queue data structure
	a new element is pushed from were whereas
rate A	First window of Rsize we need to process.
'	Scarifica With Ca

Scarined with Cam

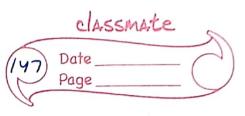


Scarined with CaM

	A A
	While processing simply bush the index of
	negative elements in the queue
	queue -> 1 2
	q.front () has some value, then avr [q.front()]
	is the answer else 0 is the answer
 *	Remove the front element of 1/p away
	Now new element insertion only if negative.
Note.	We have stored index in queue so that we can
	get to know whether it lies in the window of
	size k or not. This can be know via indexes & not.
	by storing the elements.
94	Formulae for removal of element from the
1	queue indo and condende
	$1 - 9 \cdot front() > = k$
1	12 -1 -7 8 -15 30 16 28
	Ti.
	q.front()
	4-1>=k
	3>=33 True & hence index of -1 to
	be removed from the queue.
	Code
	void solve (int aver [], int size, int k) {
	Vector <int>ans;</int>
	// Create a deque

	deque <int>q;</int>
	//Process first window of size = R for (int i=0; i < R; i++) {
	for (int i=0) (< k) ("++) {
	// Push -ve number index in deque
	if (our (i) <0) {
	q. push_back(i);
	2
	in Ser die de lancie to al led exporte i
176.	Process the remaining window for (int i = R ; i < size ; i++){
275	if (q.empty()){
19 -	
JUA	ans. bush_back(0);
	else & // Puch the frank of
sat o	else { // Push the front element in answer int temp = avr (q. front ())];
2 1 1 3 1 1	ans. bush - back (temp);
	3
	Mout of window element to be removed
8 ¢.	VIIII () >= R/L
	g.pop-front();
	5 cm c 10 m c 10
	// Push index of the -ve number if (aur [i] < 0) {
4	Cabuch look (i)
	q. push back (i);
	3
	// Process the last window
	if (q.empty()) t
	ans. bush_back (0);
	9
	else t

Scarnieu with Jam



```
int temb = our (q.front());
    ans. push - back (temb)
  1/ Print the ans vector
  for (int i=0) ixans.size();i++){
           cout << (()) << " ));
Time complexity = O(n)

Space complexity = O(R) - Can be done in constant
```

Scanned with Cam