Priority Queue	
int A(SIZE) front, rear	
create ()	
€ front ← rear = -1	
3	
insert-by-priority (int data)	
{	
if ( rear 7= SIZE-1)	
E print ( " Quene Overflow");	
return;	
3	
ig ((hont==-1) &2 (ner==-1))	
front = front + 1;	
near = near +1;	
A [ rear] = data;	
<u> </u>	
eln	
check (data);	$\dashv$
rear ++ ;	
1	$\dashv$
	$\dashv$
· Void check (int data)	
9	$\dashv$
inti, i,	
for (i=0; i <= rear; i++)	_
if (data 7 = A[i])	_
{ for ( j= reax +1; j > i j)	-
{ A[j] - A[j-1],	-1
	1
Scanned with CamScanner	

	A[i] = data;
	return;
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	Ali] = data;
	of the state of th
	void delle 1 0
	Void delete _ by - priority (int data)
	$\mathcal{M}$ ( $\mathcal{L}$
	y (( front = = -1) && (rear = = -1))
	Return;
	return;
	for (i=0; i<= rear; i++)
	{
•	y (dala == A[i])
,	lov ( is real it)
	for (; i< rear; i++)  { A[i]: A[i-1]:
	3
	Ali]= -99;
	near ,
	ig ( rear = = -1)
	front = $-1$ ;
	return;
	3
	prints (" Element not found").
	g down
3	
· 'a	

void display-queue ()
<sup>2</sup>
(11
ig ((front ==-1) & & ( rear == -1))
brints (" queue is Empty"),
brints (" Guene is Empty'); leturn;
1
for (; front <= rear; front ++)
{
heatl ("Xd" Alford)
print f (" /d", A (front));
for 1 o
front = 0;
 J.
1 y 84
 and to
The state of the s