Pseudo code → Circular Queue

current position = i

Next position = (i+1) % N

Previous Position = (i+N-1) % N

A[SIZE]

FRONT = -1

REAR = -1

```
IsFull()
{
if(front == (rear+1) % N)
return true
else
return false
}
IsEmpty()
{
if (front == -1 && rear == -1)
return true
else
return false
}

  display()
{
  if (front == -1)
  printf → Queue is Empty
else if (rear >= front)
 {
   for(i=front ; i<= rear ; i++)
     printf("%d\n", a[i]);
 }
 else
    {
      for (i = front ; i<size ; i++)
        printf("%d\n", a[i]);
      for(i=0 ; i<= rear ; i++)
        printf("%d\n", a[i]);
    }
}
```

```
Enqueue(x)
{
 if (IsFull())
 printf("Q is full")
 else if (IsEmpty())
 front ← 0 rear ← 0
 else
{
 rear ← (rear+1) % N
 }
 A[rear] = x
}

Dequeue(x)
{
if (IsEmpty())
print → Queue is empty
else if (front == rear)
{
  x = A[front];
   front = rear = -1;
 }
else
{ x = A[front];
  front = (front+1) % N;
 }
 return x;
}
```