Dequeue – Doubly Ended Queue. In this queue, we can carry out the insertion and deletion operations from either ends. This means the possible operations are

1. Insert From Left
2. Insert From Right
3. Delete From Left
4. Delete From Right

This is a special type of data structure in which we can use both the principles of insertion and deletion i.e. FIFO & LIFO.

There are 2 types of Dequeues

1. IRD – Input Restricted Dequeue. Possible operations on this queue are insert from Left, Deletion from Left & Deletion from Right.
2. ORD – Output Restricted Dequeue. Possible operations on this queue are Insert from Left, Insert from Right & Deletion from Left.

Insert From Left

#define SIZE 5

Case – 1, Queue is empty, Element Case-3, front is at position 0

is inserted @ position 0 only i i

0 1 2 3 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ~~10~~  5 | ~~20~~  10 | ~~30~~  20 | 30 |  |

Case – 2, front is not at position 0

Bring front to the left by 1

position

0 1 2 3 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ~~10~~ | 20 | 30 |  |  |

f r r

f f r To add element 5 from left, shift all

f=r=-1 elements to the right by single position

Dequeue full condition

if (pq->front == 0 && pq->rear==SIZE-1)

return 1;

return 0;

Insert From Right

Case – 1, Queue is empty, Element Case-3, rear is at position SIZE-1

is inserted @ position 0 only i i

0 1 2 3 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ~~10~~  20 | ~~20~~  30 | ~~30~~  40 | ~~40~~  50 | ~~50~~  60 |

Case – 2, rear is not at position SIZE-1

Move rear to the right by 1

position

0 1 2 3 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10 | 20 | 30 | 40 |  |

f f r

f

f r r To add element 60 from right, shift all

f=r=-1 elements to the left by single position