Circular Queues

Circular Queue

Circular Queue is a **linear data structure** in which the operations are performed based on FIFO (First In First Out) principle and the last position is connected back to the first position to make a circle.

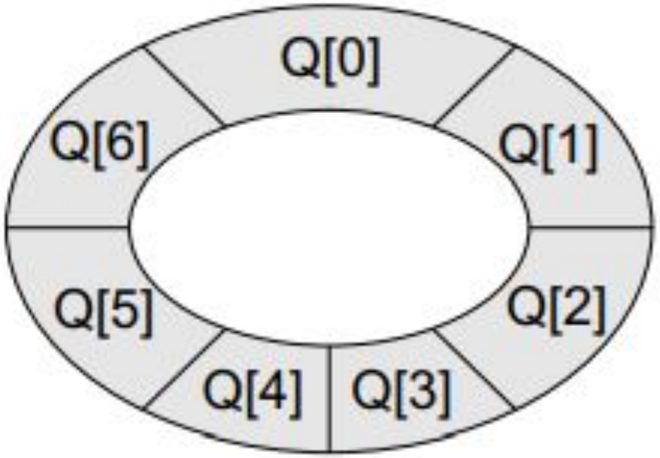
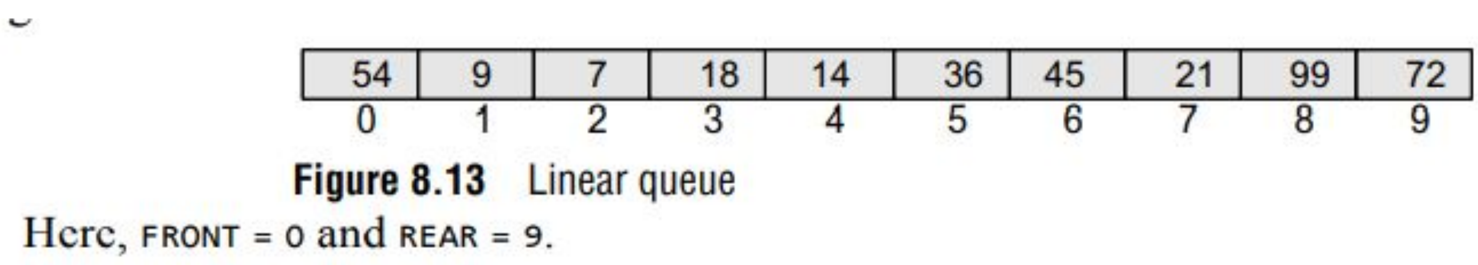


Figure 8.15 Circular queue

Circular Queue

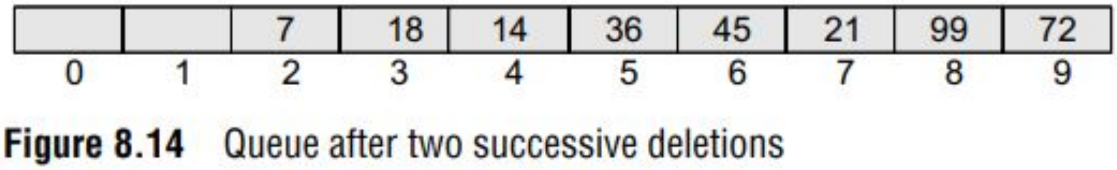
Drawbacks of normal queue:



54	9	7	18	14	36	45	21	99	72
0	1	2	3	4	5	6	7	8	9

Figure 8.13 Linear queue

Hcrc, FRONT = 0 and REAR = 9.



		7	18	14	36	45	21	99	72
0	1	2	3	4	5	6	7	8	9

Figure 8.14 Queue after two successive deletions

Circular Queue Implementation

To insert an element we now have to check for the following three conditions:

- If front = 0 and rear = MAX – 1, then the circular queue is full. Look at the queue given in Fig. 1which illustrates this point.
- If rear != MAX – 1, then rear will be incremented and the value will be inserted as illustrated in Fig. 2
- If front != 0 and rear = MAX – 1, then it means that the queue is not full. So, set rear = 0 and insert the new element there, as shown in Fig. 3

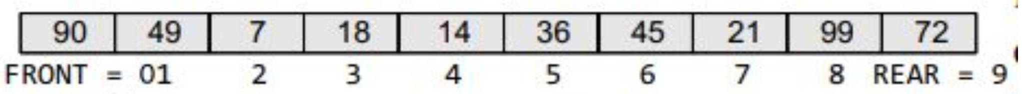
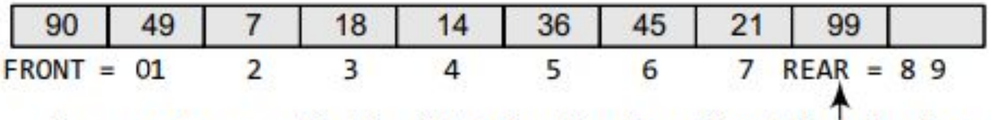
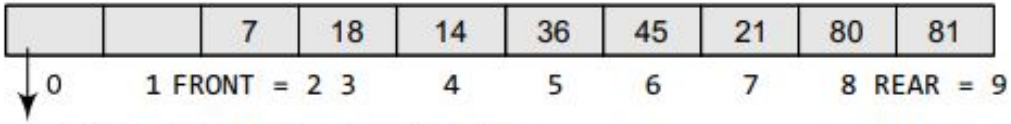


Figure 1 Full queue



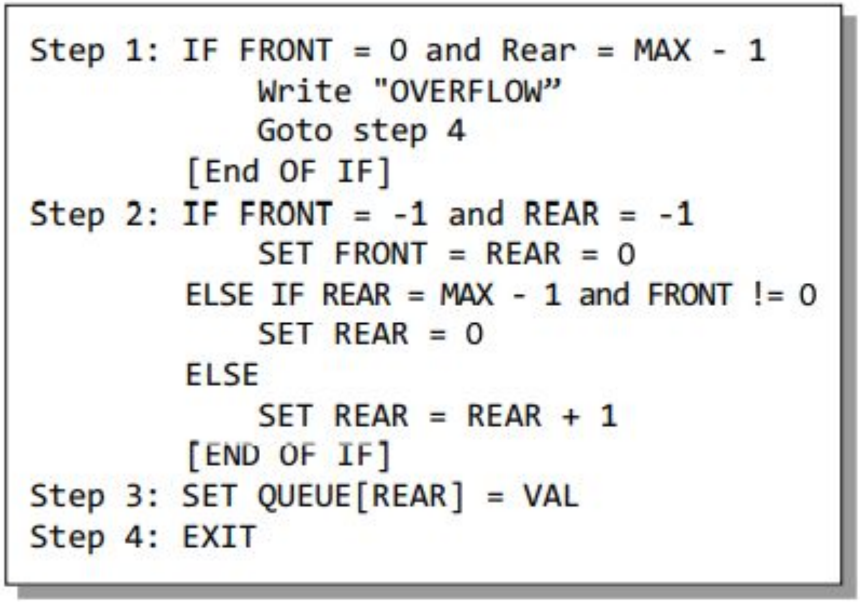
Increment rear so that it points to location 9 and insert the value here

Figure 2 Queue with vacant locations



Set REAR = 0 and insert the value here

Circular Queue Implementation



```
Step 1: IF FRONT = 0 and Rear = MAX - 1
            Write "OVERFLOW”
            Goto step 4
        [End OF IF]
Step 2: IF FRONT = -1 and REAR = -1
            SET FRONT = REAR = 0
        ELSE IF REAR = MAX - 1 and FRONT != 0
            SET REAR = 0
        ELSE
            SET REAR = REAR + 1
        [END OF IF]
Step 3: SET QUEUE[REAR] = VAL
Step 4: EXIT
```

Algorithm to insert an element in a circular queue

Circular Queue Implementation

To delete an element, again we check for three conditions.

- Look at Fig. 1. If front = –1, then there are no elements in the queue. So, an underflow condition will be reported.
- If the queue is not empty and front = rear, then after deleting the element at the front the queue becomes empty and so front and rear are set to –1. This is illustrated in Fig. 2.
- If the queue is not empty and front = MAX–1, then after deleting the element at the front, front is set to 0. This is shown in Fig. 3

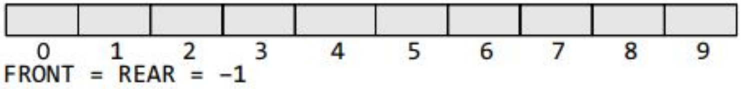
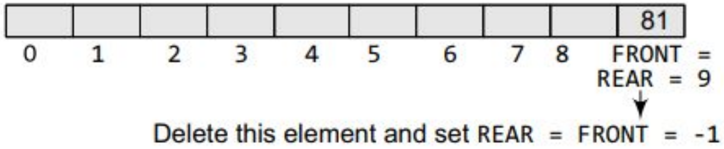
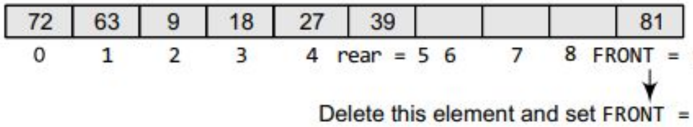


Figure 1 Empty queue



Delete this element and set REAR = FRONT = -1

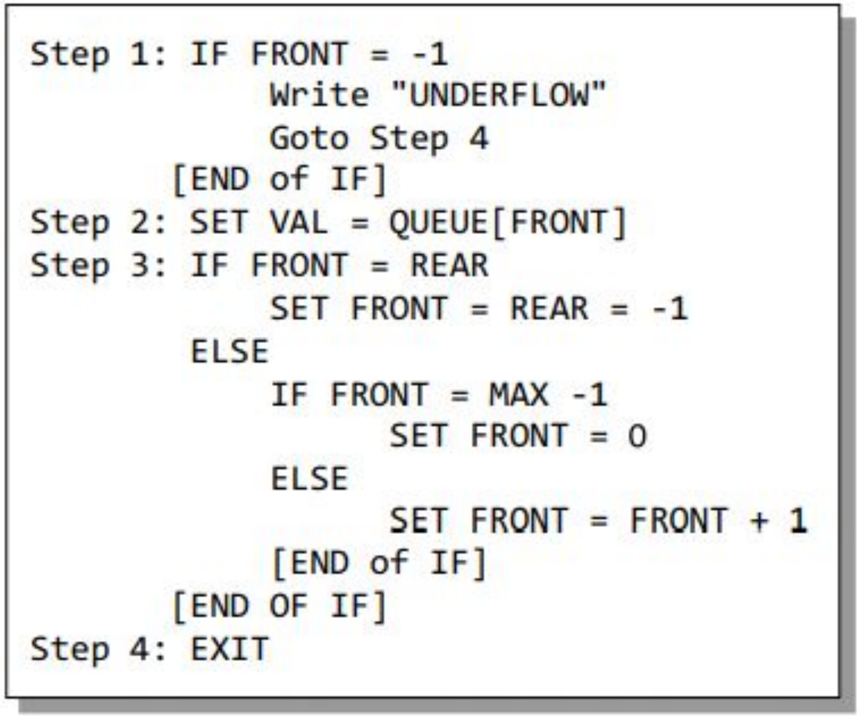
Figure 2 Queue with a single element



Delete this element and set FRONT = 0

Figure 3 Queue where FRONT = MAX–1 before deletion

Circular Queue Implementation



```
Step 1: IF FRONT = -1
             Write "UNDERFLOW"
             Goto Step 4
        [END of IF]
Step 2: SET VAL = QUEUE[FRONT]
Step 3: IF FRONT = REAR
             SET FRONT = REAR = -1
         ELSE
             IF FRONT = MAX -1
                   SET FRONT = 0
             ELSE
                   SET FRONT = FRONT + 1
             [END of IF]
        [END OF IF]
Step 4: EXIT
```

Figure 8.23 Algorithm to delete an element from a circular queue

applications of circular queue

traffic light

- Traffic light functioning is the best example for circular queues. The colors in the traffic light follow a circular pattern.