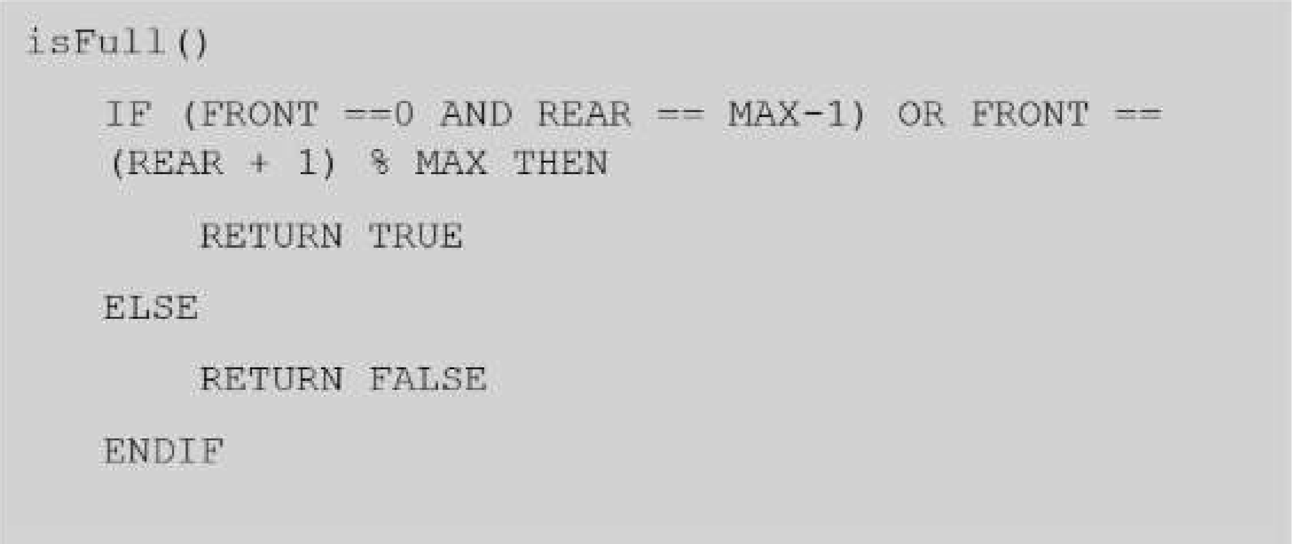
Tending to Infinity



```
isFull()
    IF (FRONT ==0 AND REAR == MAX-1) OR FRONT ==
    (REAR + 1) % MAX THEN
        RETURN TRUE
    ELSE
        RETURN FALSE
    ENDIF
    ELSE
        ITEM = QUEUE[FRONT]
        IF FRONT == REAR THEN
            FRONT = REAR = -1
        ELSE
            FRONT = (FRONT + 1) % MAX
        ENDIF
        // Optional --
        RETURN ITEM
    ENDIF
```

```
Peek()
    IF FRONT == -1 THEN
        PRINT "Queue Empty"
    ELSE
        PRINT QUEUE[FRONT]
        // Optional
        RETURN QUEUE[FRONT]
    ENDIF
```

```
isEmpty()
    IF FRONT == -1 THEN
        RETURN TRUE
    ELSE
        RETURN FALSE
    ENDIF
```

Circular Queue Algorithm

Note: The function returns the inserted element if the operation is successful, or 0 if the queue overflows or underflows. Alternatively, instead of using return statements, you can simply print the messages in both cases — the algorithm will still work correctly.

```
Enqueue(ELEMENT)
    IF (FRONT ==0 AND REAR == MAX-1) OR FRONT ==
    (REAR + 1) % MAX THEN
         PRINT "Queue Overflow"
         // Optional --
         RETURN 0;
    ELSE
         IF FRONT == -1 THEN
              FRONT = 0
         ENDIF
         REAR = (REAR + 1) % MAX
         QUEUE[REAR] = ELEMENT
         PRINT "Element Inserted" + ELEMENT
         // Optional --
         RETURN ELEMENT
    ENDIF
```

```
Dequeue()
    IF FRONT == -1 THEN
         PRINT "Queue Underflow"
         // Optional --
         RETURN 0
```