Aim: Implementation of Linear Queue using array

Theory:

Queue is an important linear data structure which stores its elements in an ordered manner.

Example:

1. People moving on an escalator. The people who got on the escalator first will be the first one to step out of it.
2. A queue is a FIFO (First-In, First-Out) data structure in which the element that is inserted first is the first one to be taken out.
3. The elements in a queue are added at one end called the rear and removed from the other one end called the front.

Array implementation of queue:

1. Queues can be easily represented using linear arrays.
2. Every queue has front and rear variables that point to the position from where deletions and insertions can be done, respectively.
3. Before inserting an element in the queue we must check for overflow conditions.
4. An overflow occurs when we try to insert an element into a queue that is already full, i.e. when rear = MAX – 1, where MAX specifies the maximum number of elements that the queue can hold.
5. Similarly, before deleting an element from the queue, we must check for underflow condition.
6. An underflow occurs when we try to delete an element from a queue that is already empty. If front = -1 and rear = -1, this means there is no element in the queue.

Algorithms:

Algorithm to insert an element in a queue

```
Step 1: IF REAR=MAX-1, then;
                Write OVERFLOW
                Goto Step 4
        [END OF IF]
Step 2: IF FRONT == -1 and REAR = -1, then
                SET FRONT = REAR = 0
        ELSE
                SET REAR = REAR + 1
        [END OF IF]
Step 3: SET QUEUE[REAR] = NUM
Step 4: Exit
```

Algorithm to delete an element from a queue

```
Step 1: IF FRONT = -1 OR FRONT > REAR, then
                Write UNDERFLOW
                Goto Step 2
        ELSE
                SET VAL = QUEUE[FRONT]
                SET FRONT = FRONT + 1
        [END OF IF]
Step 2: Exit
```