Algorithm_Queue_Insert_Using_Array()

Input: A Queue implemented using an array, say A, an element to be inserted, say ITEM and FRONT, a variable that will hold the index of the item inserted first in the queue and REAR, a variable that will hold the index of the item inserted last in the queue.

Output: ITEM successfully inserted at the REARth position of the queue otherwise suitable overflow message.

Data Structure used: An array A[L..U] where L = Lower index of the array, U = Upper index of the array and SIZE = U – L + 1

Steps:

1. Begin
2. If (REAR = U)
3. Then
4. Print “Queue overflow, ITEM can’t be pushed in stack”
5. Else
6. Set REAR = REAR + 1
7. Set A[REAR] = ITEM
8. If FRONT = L – 1
9. Then
10. Set FRONT = FRONT + 1
11. End If
12. End If
13. End

Note: The initial value of FRONT and REAR will be L – 1 when the Queue is empty

Algorithm_Queue_Delete_Using_Array()

Input: A Queue implemented using an array, say A, FRONT, a variable that will hold the index of the item inserted first in the queue and REAR, a variable that will hold the index of the item inserted last in the queue.

Output: An item, say ITEM successfully deleted from the FRONTth position of the queue otherwise suitable underflow message.

Data Structure used: An array A[L..U] where L = Lower index of the array, U = Upper index of the array and SIZE = U – L + 1

Steps:

If (FRONT = FRONT – 1)
Then
    Print “Queue underflow, no item to delete”
Else
    Set ITEM = A[FRONT]

```
        Set FRONT = FRONT + 1
        If FRONT > REAR    // Resetting the queue when all the items of the queue are
                                deleted
        Then
                Set FRONT = REAR = L – 1
        End If
        Return ITEM
End If
End
```

Note: The initial value of FRONT and REAR will be L – 1 when the Queue is empty

Algorithm_Queue_Traverse_Using_Array()

Input: A Queue implemented using an array, say A, FRONT, a variable that will hold the index of the item inserted first in the queue and REAR, a variable that will hold the index of the item inserted last in the queue.

Output: The elements of the queue are successfully traversed from the FRONT position till the REAR position otherwise suitable underflow message.

Data Structure used: An array A[L..U] where L = Lower index of the array, U = Upper index of the array and SIZE = U – L + 1

Steps:

1. If FRONT = L – 1
2. Then
3. Print “Stack underflow, no item to traverse”
4. Else
5. Set i = FRONT
6. While i ≤ U
7. Begin
8. Process (A[i]) // Process() is a procedure that processes the element being traversed in required way
9. Set i = i + 1
10. End While
11. End if
12. End
13. **Note:** The initial value of FRONT and REAR will be L – 1 when the Queue is empty

Note: The initial value of TOP will be L – 1 when the stack is empty