Aim:

Write a program to implement a Queue using Array.

Algorithm:

**Step 1** - START

**Step 2** - declare void functions enQueue(), deQueue(), and display.

**Step 3** - declare integer variable ‘size’ and array ‘queue’ of size 100

**Step 4** - initialize integer variables front is equal to -1 and rear is equal to -1.

**Step 5** – under enQueue() function and declare integer variable ‘item’.

**Step 6** - get an integer from the user to be added to the queue and store it in variable item.

**Step 7** – if rear is equal to size-1, display OVERFLOW!

**Step 8** – otherwise if front is equal to -1 and rear is equal to -1, set front is equal to 0 and rear is equal to 0.

**Step 9** – otherwise increment rear by 1 and Set queue[rear] is equal to item, in which item is inserted in queue at index ‘rear’.

**Step 10** - under deQueue() function, declare integer variable item.

**Step 11** - if front is equal to -1 or front is greater than rear, display UNDERFLOW!

**Step 12** - otherwise set item is equal to queue[front], which means element from queue at index ‘front’ is stored in variable item.

**Step 13** - If front is equal to rear, set front is equal to -1 and rear is equal to -1.

**Step 14** – otherwise increment front by 1.

**Step 15** - under display() function, declare integer variable index.

**Step 16** - if rear = -1, display Empty Queue!

**Step 17** – otherwise, introduce a for loop and initialize index is equal to rear.

**Step 18** - If the condition (index greater than or equal to front) is true, continue. otherwise exit the loop.

**Step 19** - display the element queue[index], which is the element from the queue at index ‘index’.

**Step 20** - decrement index by 1, and go to step 17.

**Step 21** – declare variable choice.

**Step 22** - get the size of the queue from the user and store the value in variable ‘size’.

**Step 23** - introduce a do-while loop having condition (choice<=4). If True, continue. Otherwise, exit the loop.

**Step 24** - get the operation to be performed on the queue from the user and store the value in variable ‘choice’.

**Step 25** - introduce a switch statement with variable ‘choice’ as its input.

**Step 26** - if choice is 1, call the enQueue() function (go to step 5) and go to Step 23.

**Step 27** - if choice is 2, call the deQueue() function (go to step 10) and go to Step 23

**Step 28** - if choice is 3, call the display() function (go to step 15) and go to step 23.

**Step 29** - if choice is 4, go to step 23.

**Step 30** - if neither of the cases were satisfied, provide a default statement that displays Invalid choice! and go to step 23.

**Step 31** - STOP

Result:

The program is successfully executed and achieved the aim of the program.

Output:

  