

# In-place and Out-place Assignment

---

Q1 :-

<i>In-place algorithms</i>	<i>Out-place algorithms</i>
➤ The algorithm does not require any additional space for operation.	➤ The algorithm requires additional space for its operation
➤ It needs only $O(1)$ extra space	➤ It needs $O(n)$ extra space
➤ Eg :- sorting algorithms such as insertion sort, selection sort, bubble sort all these algorithms require a constant amount of extra space for rearranging the elements in the input array	➤ Eg:- Merge algorithm requires $O(n)$ extra space for merging.

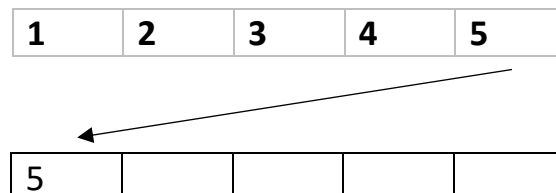
---

Q3:-

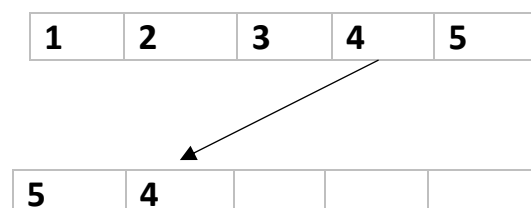
EG:- FOR REVERSING AN ARRAY

## IN CASE OF OUT-PLACE ALGORITHM :-

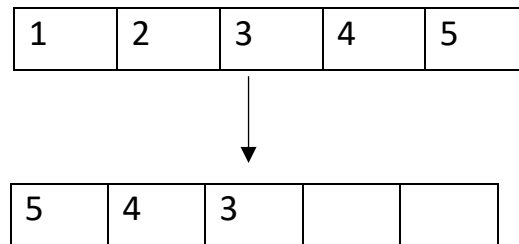
- **STEP 1:-** creation of blank array and placing last element at index 0 of blank array



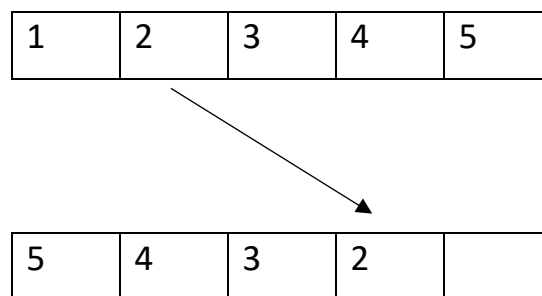
- **STEP 2:-** Place elements in reverse order



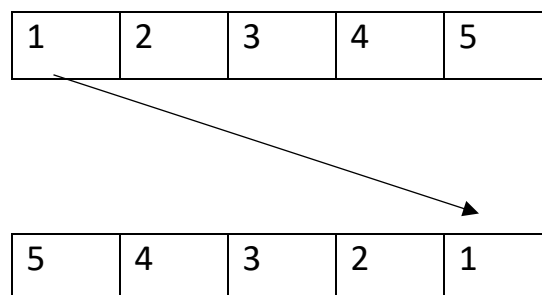
- STEP 3:-



- STEP 4:-



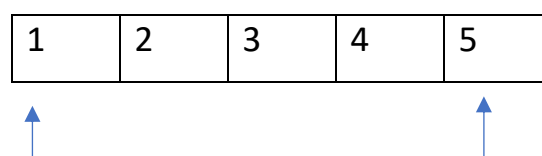
- STEP 5:-



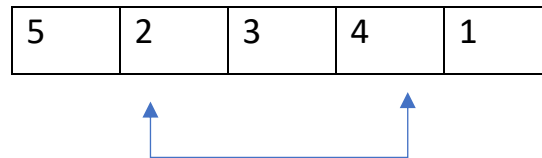
This needs  $O(n)$  extra space

## IN CASE OF IN-PLACE ALGORITHM :-

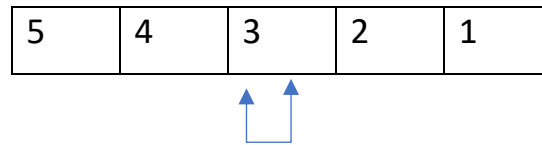
- STEP 1:- Swap element at first index and last index



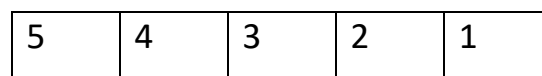
- STEP 2:- Similarly continue the process



- STEP 3:-



- STEP 4:- Final sorted Array



This needs  $O(1)$  extra space for reversing the algorithm