* Given a non-empty array of integers. Every element appears twice except for one. Find that single one. Implement it in Time: O(n) & Space: O(1).

Input: [2,2,1]

Output: 1

Ans: take the xor of all the elements, it will nullify the duplicate elements.

* [https://www.geeksforgeeks.org/***minimum-length-unsorted-subarray-sorting-which-makes-the-complete-array-sorted/***](https://www.geeksforgeeks.org/minimum-length-unsorted-subarray-sorting-which-makes-the-complete-array-sorted/) <https://www.youtube.com/watch?v=p7rnuGsau6g>.

Store one more copy of given array and then sort it using sort() function. Compare the values at corresponding indices of the 2 arrays. Calculate the first and last occurrence of the case where the corresponding elements in the arrays do not match. These are our answers. If the dimension of array is 1, just print 0 0 as it is already sorted.(Corner Case)

* count all the pairs with given sum in time O(n) space O(n).