Data Structure LAB Lab Test-1

**Max Marks:** 20 **Max Time: 4**0 min **Instructions**:

1. Submit a pdf file having code and output screenshots
2. Filename should be RollNo\_Name\_LabTest1\_DS.pdf
3. Output should be at least 3 unique test cases
4. Output Screenshot should NOT be cropped at all
5. Any two students having same output will be awarded 0 marks

# SET 1(ODD SYSTEM NO)

Q1. You are given two integer arrays nums1 and nums2, sorted in non-decreasing order, and two integers m and n, representing the number of elements in nums1 and nums2 respectively.

Merge nums1 and nums2 into a single array sorted in non-decreasing order.

The final sorted array should not be returned by the function, but instead be stored inside the array nums1. To accommodate this, nums1 has a length of m + n, where the first m elements denote the elements that should be merged, and the last n elements are set to 0 and should be ignored. nums2 has a length of n. Can you come up with an algorithm that runs in O(m + n) time?

**Input:** nums1 = [1,2,3,0,0,0], m = 3, nums2 = [2,5,6], n = 3

Output: [1,2,2,3,5,6]

Explanation: The arrays we are merging are [1,2,3] and [2,5,6].

The result of the merge is [1,2,2,3,5,6] with the underlined elements coming from nums1.

# Set 2(EVEN SYSTEM NO)

Q1. Given an integer array, find the previous smaller element for every array element. The previous smaller element of a number x is the first smaller number to the left of x in the array. Hence, for each element A[i] in the array A, find an element A[j] such that j < i and A[j] < A[i] and value of j should be as maximum as possible. If the previous smaller element doesn't in the array for any element, consider it -1.

For example,

Input: [2, 5, 3, 7, 8, 1, 9]

Output: [-1, 2, 2, 3, 7, -1, 1]