Lab 9 Tasks

Q1. The first line of input will be an integer *n.* The next line of the input will contain *n* integers. Store the integers in an array *nums*. Now reorder it such that *nums[0] < nums[1] > nums[2] < nums[3]…* and then print the array. You may assume all input has valid answer.

Example:

Input: 6 1 5 1 1 6 4	**Output:** 1 4 1 5 1 6
Input: 6 1 3 2 2 3 1	**Output:** 2 3 1 3 1 2

Q2. The input will then contain *3* separate **pairs** of lines. The first line of each pair will be an integer *n (0 < n < 11).* The second line of the pair will contain *n* integers. Put the integers in the second line in an array. Repeat the same steps for each pair of lines. [So, you will need *3* separate arrays.] Now find the **smallest** range that includes at least one number from each of the *3* arrays.

We define the range [a,b] is smaller than range [c,d] if b – a < d -c or a < c if b – a == d – c.

Example:

Input: 5 4 10 15 24 26 4 0 9 12 20 4 5 18 22 30	**Output:** 20 24

Explanation:
Array 1: [4 10 15 24 26], 24 is in range [20 24]
Array 2: [0 9 12 20], 20 is range [20 24]
Array 3: [5 18 22 30], 22 is in range [20 24]

Q3. Given a non-negative integer, you can swap any two digits of the number at most once to get the maximum valued number. Return the maximum valued number you can get

Example:

Input: 2736	**Output:** 7236
Input: 9973	**Output:** 9973

Q4. The array *num* should contain all the integers from 1 to *n* each occurring only once. However, one of the numbers in the array was replaced by a number from 1 to *n.* That means one of the numbers between 1 to *n* occurs twice while one of the numbers from 1 to *n* is missing from the array. Your task is to find which number is missing and which number occurs twice.
The first line of the input will be the integer *n.* The second line will contain *n* integers. In the output first print the number that is missing and then print the number which occurs twice.

Input: 1 2 2 4	**Output:** 3 2