CS19001: Programming and Data Structures Lab

Lab Test: 1 (ODD-PC)　　Section:15　　Date: 03-Sep-2018

Instructions:

- You have to **submit only two .c program files** (and nothing else) in the mentioned two submission links (in Moodle).
- Please obey the file-naming convention as follows: ***RollNo_MachineNo*****_LT1_Prog1.c** (for **Problem-1**) and ***RollNo_MachineNo*****_LT1_Prog2.c** (for **Problem-2**).
  [Please write your **own** Roll-Number and Machine-Number in the mentioned place-holder.]
- Submission Deadline: **03-Sep-2018, 12:00 NOON (!! STRICT !!)**

Problem-1: [Dig-Sort]

Write a C-program which –

- Takes a long (non-zero) positive integer from the user (do not enter redundant zeros in the front while entering the integer, for example, DO NOT enter 00123, instead enter 123).
- Converts the integer into the sorted digit version and stores this in a variable of long positive integer format. (Example: the sorted-digit version of the number 104120 is 001124)
- Prints the obtained result (Remember, if the entered number contains few 0s in between, you have to prefix the sorted-digit version with those many 0s also!).

However, *YOU ARE **NOT** ALLOWED TO USE ANY ARRAY*.

Execution-1**:**

```
Enter Non-Zero Positive Integer : 1031027990
Sorted-Digit Number:  0001123799
```

(Note that, the prefix 0s are exactly same in number as it was in the entered integer!)

Execution-2**:**

```
Enter Non-Zero Positive Integer : 123219876859
Sorted-Digit Number:  112235678899
```

Problem-2: [Divide-and-Rule]

Write a C-program which –

- Takes as input from the user (a) the number of elements of an array and (b) all the elements of the array in any order (can be both positive or negative but **non-zero** integers).
- Re-arranges the array elements in such a way that all the negative integers lie in the left side of the array and all the positive integers lie in the right side of the array.
- Prints the re-arranged array.

However, *YOU ARE **NOT** ALLOWED TO PERFORM ANY SORTING OPERATION, AND YOU **CANNOT*** USE ANY EXTRA ARRAY OTHER THAN THE INPUT ARRAY DECLARED.

Execution-1**:**

```
Enter Number of Elements: 5
Enter 5 Non-zero Integers: 7 -1 -5 2 9
The Re-arranged Array: -5 -1 7 2 9
```

(Note that, any output having negative elements followed by positive elements is valid!)

Execution-2**:**

```
Enter Number of Elements: 6
Enter 6 Non-zero Integers: 1 2 3 4 5 6
The Re-arranged Array: 1 2 3 4 5 6
```

(Note that, there is no change in the input array in this case!)