

# CS19001: Programming and Data Structures Lab

Lab Test: 1 (EVEN-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: [ Tour-Di-Count ]

Write a C-program which –

- Takes a (non-zero) positive integer from the user.
- Converts the integer into the *count-digit* version and stores this in a variable having long positive integer format. (Example: if you count the distinct digits in the number 104120, it gives you, 'two 0s' <20>, 'two 1s' <21>, 'one 2' <12> and 'one 4' <14>. So, the count-digit version outputs 20211214, which is obtained by writing all the mentioned relationship of digits and their counts together in the form <count digit>)
- Prints the obtained result. (Remember, the count-digit version of the number will reveal digit countings in sorted way. Ex: If input is 2001, the correct output is 201112, NOT 122011)

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

### Execution-1:

```
Enter Non-Zero Positive Integer : 1031027990
Count-Digit Number: 302112131729
```

### Execution-2:

```
Enter Non-Zero Positive Integer : 1319876899
Count-Digit Number: 211316172839
```

---

## Problem-2: [ á-la-Carte ]

Write a C-program which –

- Takes as input from the user (a) the number of elements of an array and (b) all the integer elements of the array entered in sorted (ascending) order. Moreover, the user may enter duplicates (as many as (s)he wants) but *for only one* element.
- Re-builds the array elements so that the duplicates are eliminated. (Example: if the input array is [ 1 2 2 2 3 4 5 ], then the re-built array becomes [ 1 2 3 4 5 x x ] after removing duplicates, where x represents the entries that are not meaningful anymore in the array).
- Prints this re-built array only upto meaningful elements.

However, **NO EXTRA ARRAY IS ALLOWED OTHER THAN THE INPUT ARRAY DECLARED.**

### Execution-1:

```
Enter Number of Elements: 11
Enter 11 Ordered Integers: 2 8 15 23 23 23 23 26 29 30 32
The Re-built Array: 2 8 15 23 26 29 30 32
```

(Note that, 3 duplicates of 23 are removed and array is printed only upto meaningful elements!)

### Execution-2:

```
Enter Number of Elements: 6
Enter 6 Ordered Integers: -1 0 2 4 6 8
The Re-built Array: -1 0 2 4 6 8
```

(Note that, no duplicate here and hence the same array gets printed!)

---