CS1303: Introduction to Programming
Assignment 8:
Array Operations

Submission Deadline:
Monday, 4th November 2019, 10:00 PM

Problem Statement: Use one-dimensional integer arrays to solve the following problem. Read in two Sets of numbers, each having 10 integer numbers. Input array should not have duplicate elements. **In case of any duplicate elements entered by a user, your program should ask the user to enter a unique element.** For more clarity please refer to the Sample test cases.

Then you have to develop following functions:

1) Union of Sets
2) Intersection of Set
3) Computation: First and Second Largest element in both Union and Intersection set.

1) **Union of Sets:** After reading all values, develop a function which display and store all the **unique elements** in the collection of both sets. Store the union of two sets in the new set.

2) **Intersection of Sets:** After reading all values, develop a function which display and store the **unique elements** common to both sets. Store the Intersection of two sets in the new set.

3) **Computation:** Once you find the union and intersection of two sets, next you have to find the first and second largest element in both **Union** and **Intersection** set.

Input:

1. Array1: a set having 10 unique elements.
2. Array2: a set having 10 unique elements.

Output:

1. Union Set of two input arrays (Array1 $\cup$ Array2).
2. Intersection Set of two input arrays (Array1 $\cap$ Array2).
3. First and second largest element in Union Set.
4. First and second largest element in Intersection Set.

Sample Test Cases:

1) **INPUT:**

Array1

10	4	-5	15	20	11	5	55	16	21

Array2

15	-14	3	11	35	10	1	5	16	0

OUTPUT:

Union: 10, 4, -5, 15, 20, 11, 5, 55, 16, 21, -14, 3, 35, 1, 0

Intersection: 10, 15, 11, 5, 16

Union Computation:
First Largest Element = 55
Second Largest Element = 35

Intersection Computation:
First Largest Element = 16
Second Largest Element = 15

2) **INPUT:**

Array1

3	42	-55	25	26	-11	50	54	13	21

Array2

15	-14	30	11	35	6	-1	36	16	46

OUTPUT:

Union: 3, 42, -55, 25, 26, -11, 50, 54, 13, 21, 15, -14, 30, 11, 35, 6, -1, 36, 16, 46

Intersection: NULL

Union Computation:
First Largest Element = 54,
Second Largest Element = 50

Intersection Computation:
First Largest Element = NULL
Second Largest Element = NULL

Notes:

1. In case if Intersection array is empty just print "NULL".
2. Input array should not have duplicate elements. In case of any duplicate element entered by a user, your program should ask to enter a unique element.
3. For whatever not mentioned in the problem statement you are free to take design decision, but mention that in the Design.txt file.

Submission Details:
Please submit the following information:

- **Source Code:** Your source program. The name of your file should be in this format: **Array-roll no.c** where you replace "roll no" with your roll number.

- **Readme.txt:** In this file, you should explain how to compile and run your program. The name of your file should be in this format: **Array-Readme-roll no.txt** where you replace "roll no" with your roll number.

- **Design.txt:** In this file, you explain the design of your program (control flow of your program). Your objective should be such that the TA reading this file should easily understand the working of your program. Please add details about how you have handled corner cases - i.e. for what inputs you have printed "Error". The name of your file should be in this format: **Array-Design-roll no.txt** where you replace "roll no" with your roll number.

Zip all these files and name it as **Array-roll no.zip**. **Please follow the naming convention strictly. Otherwise, your program will not be evaluated.** Then, submit it on google classroom for this assignment by the above-mentioned deadline.

Plagiarism policy: If we find a case of plagiarism in your assignment (i.e. copying of code from each other, in part or whole), you will be awarded **zero marks**. **Note** that we will not distinguish between a person who has copied, or has allowed his/her code to be copied; both will be equally awarded **zero** marks for the submission. Follow the link below for more information about plagiarism policy:
https://cse.iith.ac.in/academics/plagiarism-policy.html

Evaluation Policy:
The TAs will use the following evaluation policy:

- Design: 30%
- Execution: 60%
- Indentation and Documentation (with comments): 10%

Late Submission Penalty:
For each day after the deadline, your submission will be penalized by 10 marks.