Optional Tasks (16-21) [Ungraded]

Task 16

Write a Cpp program that reads 7 numbers into an Array and prints the **second largest number** and its location or index position on the Array. [You are not allowed to use the max(), sort(), sorted() function here]

===

Sample Input:
7, 13, 2, 10, 6, -11, 0

Sample Output:
My Array: [7, 13, 2, 10, 6, -11, 0]
Second largest number in the Array is 10 which was found at index 3.

Task 17

Write a Cpp program that reads 5 numbers into an Array and prints the smallest and largest number and their locations in the Array. [You are not allowed to use the max(), min(), sort(), sorted() functions here]

Hint: You may assume the first input to be the largest value initially and the largest value's location to be 0. Similarly, you can assume the first input to be the smallest value initially and the smallest value's location to be 0.

Note: You may need to be careful while printing the output. Depending on your code, you might need data conversion.

===

Sample Input:
7, 13, -5, 10, 6

Sample Output:
My Array: [7, 13, -5, 10, 6]
Smallest number in the Array is -5 which was found at index 2
Largest number in the Array is 13 which was found at index 1

Task 18

Write a Cpp program that takes two Arrays as an input from the user. Then print a new Array with the **common elements** of both the input Arrays.

Hint: You may need to create a third Array to store the results. You can use membership operators (in, not in) to make sure similar elements are added.

===

Sample Input 1:
A, B, C, D

C, E, F, B

Sample Output 1:
['C', 'B']

===

Sample Input 2:
1, 3, A, H, P

A, G, 1, P, O

Sample Output 2:
['1', 'A', 'P']

Task 19

Assume, you have been given two Arrays. [Your program should work for any two given Arrays; make changes to the Arrays below and check whether your program works correctly]

Array_one = [1, 2, 2, 4, 5, 5, 7, 99, 200, 303, 70]
Array_two = [1, 1, 2, 3, 3, 3, 4, 5, 200, 500, -5]

Write a Cpp program that creates a new Array with all the **unique elements** of both the given Arrays. **You need to make sure that there are no duplicates in the resulting Array.** Finally, print the updated Array.

Hint: You may create a third Array to store the results. You can use membership operators (in, not in) to make sure no duplicates are added.

===

Output for the above two Arrays: [1, 2, 4, 5, 7, 99, 200, 303, 70, 3, 500, -5]

Task 20

Write a Cpp program to take an Array as a string input from the user and print it back to the user as an Array. Please look at the examples below for clarification.

[Must use string split() and strip() functions]

==

Sample Input 1:
'1,   2 ,   3, 50, 4'

Sample Output 1:
Original data: 1,   2 ,   3, 50, 4
After removing square brackets: 1,   2 ,   3, 50, 4

```
Numbers in string format with extra white spaces: ['1', '       2' ,'            3', '50', '4']
Final data (numbers in Array format): [1, 2, 3, 50, 4]
```

==

Sample Input 2: '[1,   2   ,   3, 50, 4]'

Sample Output 2:

```
Original data: [1,      2   ,            3, 50, 4]
After removing square brackets: 1,      2   ,            3, 50, 4
Numbers in string format with extra white spaces: ['1', '      2   ', '            3', '50', '4']
Final data (numbers in Array format): [1, 2, 3, 50, 4]
```

==

Sample Input 3:

```
"      [1,   2   ,         3,   50,   4]         "
```

Sample Output 3:

```
Original data:       [1,   2   ,         3,   50,   4]
After removing square brackets: 1,   2   ,         3,   50,   4
Numbers in string format with extra white spaces: [' 1', '   2   ', '         3', '   50', '   4']
Final data (numbers in Array format): [1, 2, 3, 50, 4]
```

Task 21

Write a Cpp program that takes a single string as an input from the user where few numbers are separated by commas. Now, make an Array with the numbers of the given string. Then your task is to remove multiple occurrences of any number and then finally print the Array **without any duplicate values**.

Hint (1): For obtaining the numbers from the string, use split(). For cleaning the data, use strip().

Hint (2): You may create a third Array to store the results. You can use membership operators (in, not in) to make sure no duplicates are added.

===

Sample Input 1:

```
0, 0, 1, 2, 3, 4,      4, 5, 6, 6, 6,      7, 8, 9, 4,         4
```

Sample Output 1:

```
Given numbers in Array: [0, 0, 1, 2, 3, 4, 4, 5, 6, 6, 6, 7, 8, 9, 4, 4]
Array without any dupliacte values: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

===

Sample Input 2:

```
7, 7, 7, 1,           0, 3, 3,   55, 9
```

Sample Output 2:

```
Given numbers in Array: [7, 7, 7, 1, 0, 3, 3, 55, 9]
Array without any dupliacte values: [7, 1, 0, 3, 55, 9]
```