CS 161B: Programming and Problem Solving II

Assignment 4: Sets of Numbers

Academic Integrity

You may NOT, under any circumstances, begin a programming assignment by looking for completed code on StackOverflow or Chegg or any such website, which you can claim as your own. Please check out the Student Code of Conduct at PCC.

The only way to learn to code is to do it yourself. The assignments will be built from examples during the lectures, so ask for clarification during class if something seems confusing. If you start with code from another source and just change the variable names or other content to make it look original, you will receive a zero on the assignment.

I may ask you to explain your assignment verbally. If you cannot satisfactorily explain what your code does, and answer questions about why you wrote it in a particular way, then you should also expect a zero.

Children learn how to sort items in order at an early age. They learn how to move items down to make room for an item being inserted into the middle of their puzzle. The steps are the same when shifting items in an array!

Purpose

In this assignment, you will write a program that uses arrays to represent sets of positive integers and manipulates those arrays.

You will:

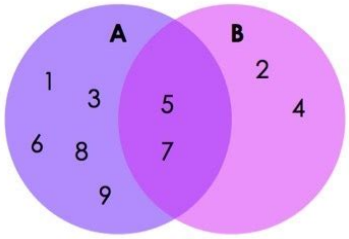
1. Shuffle two user-entered set of numbers in ascending order into their respective arrays
2. Count the number of elements in the array and insert that number as the zeroth element
3. Search the two organized arrays for intersecting elements (numbers that occur in both arrays)
4. Delete the matching (intersecting) elements found in step 3 from the first array only and update count

After completing this assignment you will be able to:

- Shift elements in an array to insert or delete elements
- Use a for loop to manipulate the array without going out of bounds
- Pass an array and the size to a function and manipulate the array

Important Information

- A set of numbers is a collection of numbers, called elements. The set can be either a finite collection or an infinite collection of numbers. One way of denoting a set, called roster notation, is to use "{" and "}", with the elements separated by commas; for instance, the set {3, 31, 7, 5} contains the elements 3, 31, 7, and 5.
    - The zeroth element (the element with array index 0) of each array will indicate the number of elements in the set. For example in the set {3, 31, 7, 5} the zeroth element will be “3” indicating there are 3 total elements in the set.
- “Intersecting elements” are numbers that occur in both sets. For example: if Set A={1, 3, 5, 6, 8, 7, 9} and Set B={5, 2, 4, 7}, then the intersecting elements are {5, 7}.



- Both arrays can hold a maximum number of 20 elements, CAP = 20. Set an integer constant for CAP
- You may not use any more temporary arrays to help you solve this problem. (But you may declare as many simple variables as you like, such as ints.) **You also may not use any other data structures or complex types such as strings, or other data**

structures such as Vector. Use only the concepts and functions we have learned so far.

- **You may not use any Sorting algorithms.**
- Your program must have function prototypes. Place the prototypes for your functions globally, after your #includes. All functions must be implemented after main().
- Use constants where appropriate, for example - the capacity of the array.
- Try not to have any redundant code (repeated code) in your program. That is the purpose of functions.
- See the Sample Run in the Criteria for Success section below.

Task

Write functions to do the following. Each function should do **ONLY** what it is supposed to do, and no more unless specified. Ater writing each function, call the function in main() to test it before writing the next function.

Your **`main()`** function should:

a. Declare 2 int arrays set1 and set2 of capacity CAP (integer constant for CAP = 20)
b. Call the **`readInput(int list[])`** function 2 times to populate the sets.
c. Call **`findIntersect(int set1[], int set2[]`**, that finds the common numbers and calls the **`deleteNum(int list[], int value)`** function to delete each number.

- ❏ Write welcome message (void function)
- ❏ Write a function called **`readInput(int list[])`** that does the following:
  - ❏ Reads an integer from the user and validates it to make sure it is a positive integer or -1. -1 ends the input. Use a loop to repeat until you get valid data from the user.

    Hint: *Write a* `readInt()` *function to do data validation. My sampleA04.cpp has this. Use char arrays instead of strings.*
  - ❏ The number of elements in the set should be entered into index 0 of the array, and make sure this number is always less than the capacity of the array, 20.
- ❏ Use the same function, send another int array to it and fill it with positive numbers in the same way. The number of elements in the set should be entered into index 0 of the array, and make sure this number is always less than the capacity of the array, 20.
- ❏ As you insert the elements in the array, order the numbers in ascending order. For help doing this see Section 11.14 in zyBooks (Figure 11.14.2: How to insert sorted into an array without using any sorting algorithms) Please also see the sample run in the Criteria for Success section below.
- ❏ Write a function called **`findIntersect(int set1[], int set2[])`**, that
  - ❏ finds the intersecting elements of the 2 arrays
  - ❏ Calls a function **`deleteNum(int list[], int value)`** which removes all the intersecting numbers from set 1 and updates the zeroth element. Set 2

remains unchanged. My sampleA04.cpp shows you how to delete numbers from an array. ***Hint:*** *Remember, the array is always passed by reference, and the updated size after deleting must be stored in the zeroth element of the array.*

Criteria for Success

- ❏ Look at the sample run below. Use the function prototypes you will need for the various options (your list may include other options):

```
Welcome to my Set of Numbers program!!

Enter numbers for set 1 on separate lines (-1 to end):
>> 6
>> 4
>> 12
>> 3
>> -1

Your set 1 with 4 numbers ordered:
3 4 6 12

Enter numbers for set 2 on separate lines (-1 to end):
>> 6
>> 4
>> -1

Your set 2 with 2 numbers ordered:
4 6

The new sets are:

Set1: 3 12
Set2: 4 6

Thank you for checking out my Set of Numbers program!
```

```
Welcome to my Set of Numbers program!!

Enter numbers for set 1 on separate lines (-1 to end):
>> 9
>> 4
>> 12
>> 3
>> 16
>> d
```

```
Invalid number! Please try again!
>> 21
>> -1

Your set 1 with 6 numbers ordered:
3 4 9 12 16 21

Enter numbers for set 2 on separate lines (-1 to end):
>> 12
>> -9
Invalid number! Please try again!
>> 9
>> 16
>> -1

Your set 1 with 3 numbers ordered:
9 12 16

The new sets are:

Set1: 3 4 21
Set2: 9 12 16

Thank you for checking out my Set of Numbers program!
```

- ❏ Before you get started:
  - ❏ Check out Sample Assignment A04 - Algorithmic Design document
  - ❏ Check out Sample Assignment A04 - sampleA04.cpp
- ❏ Complete zyBooks section **Chapter 11: Sections 11.12 to 11.14 CS 161B Arrays** activities, and **zyLabs 11.17 and 11.18.**
- ❏ You may use any IDE to write your code. But to be successful in future CS classes use the Linux server as much as you can.
- ❏ Please bookmark the PCC Linux and Vim Manual - this will become a frequently used reference!
- ❏ Complete all sections of your Algorithmic Design Document.
- ❏ **Follow these Coding Construct Requirements:**
  - ❏ Do not use any vectors or containers for this program. Use only the concepts we have learned so far. Use only the 2 arrays for the 2 sets. No more extra arrays are needed.
  - ❏ The first element in each array must hold the number of elements.

- ❏ Do not use any goto statements. You may not use any breaks or return statements inside any loops - you are allowed to use breaks inside a switch statement.
- ❏ All input data must be validated. For example, you should not accept any characters for numerical data types.
- ❏ Must not use any sorting algorithms, the numbers should be inserted in the right order.
- ❏ Must use all functions mentioned under Task - there should be no code redundancy.
- ❏ Print a welcome and goodbye message.

- ❏ Include **pseudocode** in part d of the design document.
- ❏ Please open and compare your work with the grading rubric before submitting.
- ❏ Remember to follow all style guidelines.
- ❏ Download your Algorithmic Design Document as a PDF (File -> Download -> PDF), rename it to `a04.pdf`, and upload it to the D2L assignment by **Wednesday**.
- ❏ Upload your `a04.cpp` C++ source file to the D2L assignment by **Sunday**.
- ❏ Do your own work. Consult the syllabus for more information about academic integrity.

Additional Support

- ❏ Before you get started:
  - ❏ Check out Sample Assignment A04 - Algorithmic Design document
  - ❏ Check out Sample Assignment A04 - sampleA04.cpp
- ❏ Post a question for the instructor in the Ask Questions! area of the Course Lobby.