Laboratory 10 Activity Instructions

1. Download or copy the **`Lab10Activity.cpp`** file from WebCampus. Implement the following activities, but compile and run the program after each step

2. This program is going to take all the even numbers in a given array and place them at the beginning of the array without affecting any of the subsequent numbers.

3. The following exercise assumes that your line numbering is turned on in your editor. If it is not, you will need to go to "Tools", "Editor Options", then go to the "Display" tab, and then select the "Line Numbers" box. The line numbers for the following are assumed to start at line 1 so you are better off not to check the "Start at Zero" box.

4. You will need a function to load sequential numbers into an array. The function loadNumbers does this. Place it at line 27 in your program.

5. Make sure you compile after this and every line you add to the program.

6. You will need to loop across the loaded list so you will use a "for" loop. Note that you will initialize your "from" index (i.e., the index you will be using to get data from) at zero, and you will also initialize your "to" index (i.e., the index of the same array where you will load even numbers) to zero. Look at the "for" loop code to see how this is done.

7. Inside the loop, you will need to check the next value for an even number; place the following at line 33: **`if( intArray[ fromIndex ] % 2 == 0 )`**

8. Inside the if statement, you will need to assign the value from the present element to the appropriate new list element. Place the following at line 36:
   `intArray[ toIndex ] = intArray[ fromIndex ];`

9. You will need to call the **`displayNumbers`** function to display the whole array as it is presently loaded. Place the following at line 49:
   `displayNumbers( intArray, numOriginalNums, "Original List" );`

10. You will then need to call the **`displayNumbers`** function to display only the part of the array that has guaranteed to have even numbers. Place the following at line 54:
    `displayNumbers( intArray, numEvenNums, "New Even List" );`

11. Note that the **`displayNumbers`** is used to display the array for both conditions. This is a generalizable function that can be used to display any array, given the array and the number of items to display.

12. Also note that the number of even numbers found is also the same as the **`toIndex`** variable after it has been used to assign the value and then incremented

13. You will also need to set up the functions to conduct some of the operations. The **`loadNumbers`** function loads an array from index 0 on up, but it loads it with numbers starting at 1. Place the following at line 85:
    ```
    for( index = 0, number = 1; number <= numberToLoad; index++, number++ )
    ```

14. If you have kept up with compiling with each of your lines, your program should now compile and run. Go ahead with running the program.

15. As the program runs, you will see changes to BOTH array displays. Remember that they are the same array but they have different sizes. Make sure you understand what is happening with the array in this program but if you don't understand something, ask your TA.

16. Note that the array actually has a capacity of 25. You can create as many as 25 values if you change the **`numOriginalItems`** quantity on line 22. You should be very clear that the capacity of the array (i.e., the number of items you MIGHT store in the array) is different from the size of the array (i.e., the number of items you are ACTUALLY storing in the array).

Again, if you have any problems, ask your TA for help