**Note: It would be helpful if you complete the zyBook activity for chapter 3 of the book before you start the assignment. Remember that the zyBook activity is mandatory and these activities make for 20 points of your grade.**

**You need to go to the book site to complete the activities**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Note: It would be helpful if you complete the zyBook activity for chapter 3 of the book before you start the assignment. Remember that the zyBook activity is mandatory and these activities make for 20 points of your grade.**

**You need to go to the book site to complete the activities**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Part 1, Write and Test an Array Class

In a CPP named Array.cpp, write a data structures class. The resulting class can be used in any program in place of a C++ array, in case you want the advantages of range safety and built-in size tracking.

**Requirements.** Write a Array.cpp with **class Array** defined and *fully tested* in it. Write the public interface exactly as specified below -- do not add to, or change the public interface as specified.

1. Write the class for an array of 100 *int*values, all initialized to the default int value, zero. That means you need a *main constructor.*
2. Include a square bracket getter and a setter, both with index range-checking, returning whatever value you wish, if out of range.
3. Include a getter named **Array::capacity( )**to return the data structure's capacity.

Part 2, Write an Array Application

Modify your Array.cpp Either copy/paste your code from part 1 *or* save the part 1 CPP with a new name and edit it.

The app lets its user enter as many values as they like, and when that process is completed, lets the user look up values by matching index.

In a loop, the app should prompting the user to enter a *pair of numbers* on the same line: a whole number *index* and its corresponding whole-number *value*. Do *not*validate input in the app, because your template should handle out-of-range indexes, and it should allow overwriting an already-entered index. Quit the loop when an *uppercase* or *lowercase* Q is entered for *either* the index or the value. Indexes can be entered in any order -- they don't have to start with zero and go up by one thereafter. It's whatever the user enters.

Your app should keep track of which indexes got entered. Use a second **Array**for that, with whatever integer codes you wish to distinguish used indexes from those never used.

After all data entry is complete, the app should:

1. output how many (unique) indexes got entered,
2. output the list of all used indexes and their values, per the example below, and
3. implement an event-controlled loop that prompts for an index value and outputs whether the index is in use or not, and if in use, what is the value stored for that index. Loop until the user elects to stop by entering uppercase or lowercase Q.

Here's a sample of how this should work (user input in blue):

Input an index and a value [Q to quit]: 33 12

Input an index and a value [Q to quit]: 4 100

Input an index and a value [Q to quit]: 5 300

Input an index and a value [Q to quit]: x 17

Input an index and a value [Q to quit]: 33 120

Input an index and a value [Q to quit]: -1 234  
Input an index and a value [Q to quit]: 2000 -999

Input an index and a value [Q to quit]: q

You stored this many values: 4

The index-value pairs are:

  0 => 17

  4 => 100

  5 => 300

  33 => 120

Input an index for me to look up [Q to quit]: 33

Found it -- the value stored at 33 is 120  
Input an index for me to look up [Q to quit]: 35  
I didn't find it

Input an index for me to look up [Q to quit]: 1000

I didn't find it

Input an index for me to look up [Q to quit]: Q

Design the prompts and the output formatting as you like.

Submit 3 Files

Submit Array.cpp and main.cpp for grading. Also, please submit your code on a word document followed by pictures of your app output