Assignment 3

Due: 10/20 @ 11:59 pm

50 points

*Learning Outcomes:*

1. Master the use of functions in C++.
2. Utilize call-by-value and call-by-reference implementation of functions.
3. Demonstrate mastery of pointer arrays.
4. Become familiar with dynamic memory allocation using pointers.

*Description:*

Develop a program that mimics some of the functionalities of an ArrayList in Java. Your program should maintain a pointer array of doubles and be able to perform the following functions:

1. (5 pts) insert(int index, double num, double \*&arr, int &size)
   1. Adds an element (num) to the array (arr) at a given position (index) and updates the size.
   2. You may allow the user to add to the immediate end of the array (at position n+1 for an array of n elements) but not past. You should print an error message if they try to print beyond these bounds.
2. (5 pts) remove(int index, double \*&arr, int &size)
   1. Removes an element from the array (arr) at a given position (index) and updates the size.
   2. If index is out of the bounds of the arr then an error message should be printed.
3. (2 pts) get(int index, double \*arr, int size)
   1. Returns the element at the given position (index).
   2. Should check if the index given is outside the bounds of the array. If it is out of bounds an error message should be printed.
4. (2 pts) clear(double \*&arr, int &size)
   1. Clears all elements of the array (arr) and updates the size (size) to be 0.
5. (2 pts) find(double num, double \*arr, int size)
   1. Returns the first index in which a given element (num) is found in the array (arr). If not found -1 is returned.
6. (3 pts) equals(double \*arr1, int size1, double \*arr2, int size2)
   1. Returns true if the contents of the two arrays are equal and false if they are not equal.
7. (2 pts) init(double \*arr, int size)
   1. Populates the elements of the array (arr) with input from the user (or via file redirection).
8. (2 pts) print(double \*arr, int size)
   1. Prints the elements of the array.

*Additional Specifications:*

* (-10 pts) Your program **should not** use any pre-existing classes such as string or vector classes!
* **(-5 pts) NO GLOBAL VARIABLES!**
* (5 pts) Your program should consist of a header that contains the following information:
  + Firstname and lastname of the programmer.
  + Date and time of the program completion.
  + A brief description of the program function.
  + Input requirements and format.
  + Output of the program.
  + Any additional needed comments (e.g. related to compilation, execution or other requirements).
  + Any information related to the licensing agreement.
* (2 pts) Each function needs to be properly commented.
  + Your comments need to include a description of the function.
  + Description of the inputs.
  + Description of the output.
  + Any additional notes assisting future programmers to comprehend the complex portions of your functions.
* (5 pts) Make sure your program compiles and runs on one of the Linux machines in the Linux lab before you submit.
* (5 pts) Your program should consist of two source files: Array.h and Array.cpp. They must be named exactly as indicated.
* Submit both the .h and .cpp files.
* Put the two files in a folder and **ZIP** and **NAME** your submission: Assignment3\_<last\_name>.zip (or tar.gz) (replace <last\_name> with your own last name.) **DO NOT SUBMIT A RAR FILE!** Stick with either zip or tar (these are easily decompressed on linux).

*Example Outputs:*

