Assignment 4

Due: 11/2 @ 11:59 pm
50 points

Learning Outcomes:

1. Become familiar with OOP in C++.
2. Demonstrate mastery of pointer arrays.
3. Become familiar with dynamic memory allocation using pointers.

Description:

Develop an object that mimics some of the functionalities of String class in Java. Your object should maintain a pointer array of characters and be able to perform the following functions:

1. `Your object should be named myString with the code properly distributed between myString.h and myString.cpp`
2. `(5 pts)Your object should have two constructors:`
   a. (2 pts) A default constructor that will set the size to 0 with no content
   b. (3 pts) An alternate constructor that will receive *size* and *an array of characters* as the initial values of the string.
3. `(5 pts) insert(int index, char value)`
   a. Adds an element (`value`) to the string at a given position (`index`).
   b. 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.
4. `(5 pts) remove(int index)`
   a. Removes an element from the array at a given position (`index`) and updates the size.
   b. If index is out of the bounds of the array then an error message should be printed.
5. `(2 pts) get(int index)`
   a. Returns the element at the given position (`index`).
   b. 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.
6. `(2 pts) clear()`
   a. Clears all elements of the array and updates the size (`size`) to be 0.
7. `(2 pts) find(char value)`
   a. Returns the first index in which a given element (`value`) is found in the array. If not found -1 is returned.
8. (3 pts) equals(myString)
   a. Returns true if the contents of the two strings are equal and false if they are not equal.
9. `(2 pts) print()`
   a. Prints the string to stdout.

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 method 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: *myString.h* and *myString.cpp*. They must be named exactly as indicated.
- Submit both of the .h and .cpp files.

Example Outputs: