LW: C-String

# Objective

* Use a C-string instead of the C++ string class.
* Work with an array that uses a sentinel value to mark the end of the array instead of tracking the size with a separate variable.

# Completion

Get 14 of 19 points when you submit to the autograder.

# Submission

Submit these files

* main.cpp
* cstring.cpp
* cstring.h

# What to do

1. [Get the starter code](https://drive.google.com/file/d/1ROFyunaJhGJp8DSn4KGRDJuvgYCLnyJB/view?usp=sharing).
2. Implement the three functions:
   1. unsigned int length(char str[]); - 5 Points
      1. Returns the length of the string
      2. Examples
         1. length("") should return 0
         2. length("fifteen symbols") should return 15
   2. unsigned int find(char str[], char character); - 5 Points
      1. Returns
         1. The index of the first occurrence of character in the string
         2. The size if the character is not found
      2. Examples
         1. find("the rain in spain", 'a') should return 5
         2. find("abcdefghijklmnoqrstuvwxyz", 'p') should return 25
   3. bool equalStr(char str1[], char str2[]); - 5 Points
      1. Returns
         1. true if the two strings are equal
         2. false if the two strings are not equal
      2. Examples
         1. equalStr("apple", "apple") should return true
         2. equalStr("apple", "orange") should return false
         3. equalStr(“Apple”, “apple”) should return false
3. Compilation and Forbidden Includes tests - 4 points
4. If you have time, uncomment the “unpredictable” code.
   1. Note that the string does not have a '\0' as its last character.
   2. Run and see what happens? Your results will vary since this is a very insecure thing to do for code. Try to explain any results you get. It is useful if different people run it. Try submitting it to Gradescope. You can see your output in the last test case that is worth 0 points.