Project Name: Project 4

Name: Marc Luzuriaga

UID: 205916585

**Notable Obstacles**

The biggest obstacle that I overcame in this project was figuring out how to do the shiftLeft function correctly. More specifically, I found shifting the array the difficult part of the project. At first, I tried deleting the value at an index position. However, this solution turned out to complicate the project. Instead, I just shifted the array once using a for loop from an index position, and I would have a second loop to shift it a number of times. This was a much better and simple solution.

Another obstacle that I overcame was figuring out how the locateMaximum function worked. At first, I tried comparing each element in each element of the array. However, that turned out to complicate things. Instead, I just used the comparison operators, < and >, to compare the elements.

**Test Data:**

1.

string empty[5] = {};

assert(locateMaximum(empty, 5) == 0);

Reason for test: I wanted to ensure that the function caught empty strings and returned the first occurrence of the maximum string if the array contained all identical strings

2.

string data2[11] = { "1", "1.", ".1", "01", ".01", "01.", "1 1", "1 ", "1,000,000", "1..0", "." };

assert(countFloatingPointValues(data2, 0) == -1);

Reason for test: I wanted to ensure that the if-else statement detecting if n <= 0 caught n=0 in this test.

3.

string data2[11] = { "1", "1.", ".1", "01", ".01", "01.", "1 1", "1 ", "1,000,000", "1..0", "." };

assert(countFloatingPointValues(data2, 11) == 6);

Reason for test: I wanted to ensure that the function counted the floating point numbers correctly. Specifically, I did not want my function to detect elements that contained a comma, space, two or more commas, or simply a “.” as valid.

4.

string data2[11] = { "1", "1.", ".1", "01", ".01", "01.", "1 1", "1 ", "1,000,000", "1..0", "." };

assert(shiftLeft(data2, 3, 4000, "aa") == 3);

Reason for test: I wanted to ensure that my function did not count the number of times it inserted the placeholder outside of the array. The function returned 3 because “aa” was only replaced 3 times.

5.

string empty[5] = {};

assert(hasNoCapitals(empty, 5) == true);

Reason for test: I wanted to ensure that my function did not identify empty strings as capital, and I wanted to ensure that passing empty strings into the hasNoCapitals function compiled correctly.