Ayan Nath

605910299

**Obstacles**

* One main obstacle that I encountered was with the countFloatingPointValues function. I was having trouble on how to determine if the given string was a floating point value with the conditions given in the project description. After lot of attempts, I realized that it would be easier to use helper functions. So, I had one helper function called hasOnlyDigits that made sure that the string only had numbers 0-9, “+”, “-”, and “.” and returned true if it met those conditions. Then I had another helper function called isFloatingPoint that made sure there was only one decimal “.”; Also, if there was a “+” or “-“ this function made sure there was only one and that it only existed at the front of the string. At the start of the isFloatingPoint helper function, it called hasOnlyDigits and made sure that condition was true as well. If hasOnlyDigits returns false, then the function isFloatingPoint would also return false. Then, in the actual countFloatingPointValues function, I looped through the array, and if a value in the array satisfies isFloatingPoint (and by extension hasOnlyDigits), I increased the count of floating point values by one. After the loop, I returned the count of floating point values.

**Test Cases:**

Below are the arrays I used as test cases (some of them were from the project description and discussion forum):

string people[5] = { "howard", "pixie", "barak", "joe", "donald" };

string folks[8] = { "samwell", "jon", "margaery", "daenerys",

"tyrion", "jon", "llewmas", "noj" };

string data[5] = { "mama", "mama", "12,", "sansa", "mama" };

string dee[5] = { "mama", "mama", "mama", "mama", "mama" };

string data2[ 4 ] = { "4.4.3.3", "+44", "-33.098", "33.098a" };

string v[5] = {"1", "3", "2", "2", "3"};

string zz[4] = {"a", "", "b", ""};

string yy[5] = {"A", "A" ,"B", "B", "C"};

string xx[5] = {"A", "A", "B", "A", "A"};

string ww[9] = {"A", "A", "C", "B", "B", "T", "D", "D", "D"};

For each array, I ran all 8 methods on them to make sure the output was correct: locateMaximum, matchingValuesTogether, hasDuplicates, majorityElement, hasReverse, findLastOccurance, countFloatingPointValues, replaceAll. Below are some examples of the function calls:

cout << locateMaximum( folks, 8 ) << endl; // tests to see if “4” is printed

cout << matchingValuesTogether( people, 5 ) << endl; // tests to see if “true” is printed

cout << matchingValuesTogether( folks, 8 ) << endl; // tests to see if “false” is printed

cout << hasDuplicates( people, 5 ) << endl; // tests to see if “false” is printed

cout << hasDuplicates( people, 0 ) << endl;// tests to see if “false” is printed

cout << hasDuplicates( folks, 8 ) << endl;// tests to see if “true” is printed

cout << majorityElement( folks, 8 ) << endl;// tests to see if “” is printed

cout << majorityElement( data, 5 ) << endl;// tests to see if “mama” is printed

cout << hasReverse( people, 5 ) << endl;// tests to see if “false” is printed

cout << hasReverse( folks, 8 ) << endl;// tests to see if “true” is printed

cout << findLastOccurrence( data, 5, "mama" ) << endl; // tests to see if “4” is printed

cout << findLastOccurrence( data, 5, "howard" ) << endl; // tests to see if “-1” is printed

cout << countFloatingPointValues( data2, 4) << endl; // tests to see if “2” is printed

cout <<countFloatingPointValues( data2, -14) << endl; // tests to see if “-1” is printed

cout << replaceAll( folks, 8, 'A', 'Z' ) << endl; // tests to see if “0” is printed

cout << replaceAll( folks, -10, 'A', 'Z' ) << endl; // tests to see if “-1” is printed

cout << replaceAll( folks, 8, 'a', 'Z' ) << endl; // tests to see if “5” is printed