Zane Koch

2/17/18

404984674

CS 31

Report Project 4

For the first function, findMinimum, I did not know how to compare strings initially. I thought that I would have to compare each character in a string but I soon realized that c++ does that automatically for me.

In the second function writing the code was fairly easy, the bigger problem f or me was testing it. This is because I could not figure out how to input a double quotation mark using an array of strings. After some searching on the web I found that you can have c++ treat any character as a normal character by using a back slash.

In the third function I did not have any problems.

In the fourth function, remove duplicate values, the hardest part was reordering the function so the empty strings came last. I did this in two steps. First I iterated over the string backwards so the empty string would already be nearer to the end. Second I wrote a loop that checked if there was any non empty characters after an empty character and if there was I switched them until there was not.

In the fifth function, replaceAll, I was able to get it to output the number of replacements made easily but I could not figure out to actually change the array at first. This was caused by me only changing local variables inside the loops I made. This was easily remedied after I figured it out by referencing the array itself inside the loops instead of those variables.

In the last function, shiftRight, I did not have any notable problems.

Test Data:

**findMinimum**

testArray[4]={“a”,”aa”,”b”,”bb”};

Because these are easily seen which is smallest by the human eye, if the loop does not output 0 it is working incorrectly.

testArray[0]={}

if this does not return -1 it is working incorrectly

**countAllPunctuation**

testArray[4]={ '.' , ',', '!', ';', ''', '-', '/', ':', '?', '"' }

If this does not return 10 it is working incorrectly. It is a good test case because it takes into account all possible punctuations at once.

testArray[4]={“bob.”,”j/o”,”he?nry”,”,”}

This test and makes sure that it can find punctuation inside of words.

testArray[0]={}

if this does not return -1 it is working incorrectly

**countFloatingPointValues:**

string data[ 4 ] = { "4.4.3.3", "44", "33.098", "33.098a" };

This tests and makes sure that it can identify when floating point values exist. It makes sure there can be no letters and it makes sure there can only be one period. It should output 2.

testArray[0]={}

if this does not return -1 it is working incorrectly

**removeDuplicatedValues:**

string names[4] = { "joe","joe","sam","jane" };

This tests and makes sure that it can both replace any duplicate with an empty string and then move the string to the end. This is all the function needs to do. It should replace joe with “” and move “” to the end. It should output 1.

testArray[0]={}

if this does not return -1 it it working incorrectly

**replaceAll:**

string word[4] = { "happy", "aaa", "BBB", "AAA" };

This should output 4 because it should replace the four lower case a’s. It should not replace the upper case a’s. It should then output a new array string word[4] = { "hzppy", "zzz", "BBB", "AAA" }; if it is called with the parameters of a and z.

testArray[0]={}

if this does not return -1 it it working incorrectly

**shiftRight:**

string nameroos[5]= { "samwell", "jon", "margaery", "daenerys", "tyrion" }; assert(shiftRight(people, 5, 3, "foo") == 3);

This should return 3 and then should return the array string nameroos[5]= { “foo”,”foo”,”foo”, "samwell", "jon",}

This is a good test case because it makes sure that the function works as it is told for non fringe cases.

string nameroos[5]= { "samwell", "jon", "margaery", "daenerys", "tyrion" }; assert(shiftRight(people, 5, 10, "foo") == 3);

This should output 5, because that is the maximum number of terms that can be replaced. It should then output the array string nameroos[5]= { "foo", "foo", "foo", "foo", "foo" };

This is a good test case because it tests the fringe case when the number of replacements is bigger than the number of items in the array.

testArray[0]={}

if this does not return -1 it it working incorrectly