1. Obstacles:

* Initially, the findMinimum function only compared two consecutive elements of the array, which caused several issues with getting the actual minimum value in the array.
* For removeDuplicatedValues, in the first stages, I wasn’t ignoring the empty strings that had already been replaced, so two empty strings would get compared and the result was inaccurate. I was also initially unable to move all the empty strings to the back of the array and the unique values to the front.
* For countAllPunctuation, I was not able to add single and double apostrophes as punctuation to be checked, so the result was inaccurate. However, after using an escape, the function worked as expected.

1. Test data:

|  |  |  |
| --- | --- | --- |
| **TEST DATA** | **FUNCTION CALL** | **RESULT** |
| string a[6] = {"beta", "gamma", "gamma", "alpha", "beta", "delta"}; | findMinimum(a, 4)  Tests for a unique minimum value | 3 |
| string b[6] = { "delta", "gamma", "beta", "alpha", "beta", "alpha" }; | findMinimum(b, 6)  Tests for the result when there are multiple minimum values | 5 |
| string b[6] = { "delta", "gamma", "beta", "alpha", "beta", "alpha" }; | findMinimum(b, -9)  Tests for the result when the inputted size is negative | -1 |
| string c[3] = {"a", "b", "aa"}; | findMinimum(c, 3)  Tests for minimum value in strings of different length with similar characters | 0 |
| string data[ 4 ] = { "howard-", "ucla.", "howard", "ucla." }; | countAllPunctuation( data, 4 )  Tests when some elements of the array have punctuation marks and others don’t | 3 |
| string data[ 4 ] = { "howard-", "ucla.", "howard", "ucla." }; | countAllPunctuation(data, -14)  Tests the result when the size of the array inputted is lesser than or equal to zero | -1 |
| string punct2[3] = {"a", "b", "c"}; | countAllPunctuation(punct2, 3)  Tests the result when none of the elements of the array have punctuation marks | 0 |
| string data1[ 4 ] = { "4.4.3.3", "44", "33.09e8", "33.098" }; | countFloatingPointValues(data1, 4)  Tests the result for some valid floating point elements of the array | 2 |
| string data1[ 4 ] = { "4.4.3.3", "44", "33.09e8", "33.098" }; | countFloatingPointValues(data1, 3)  Tests for the result when the size inputted is lesser than the number of elements in the entire array | 1 |
| string data1[ 4 ] = { "4.4.3.3", "44", "33.09e8", "33.098" }; | countFloatingPointValues(data1, -9)  Tests the result when the inputted size is lesser than or equal to zero | -1 |
| string floatingData[ 3 ] = {"4.4.3.3", "44,", "33.09a"}; | countFloatingPointValues(floatingData, 3)  Tests the result when none of the elements of the array are valid | 0 |
| string people[ 5 ] = { "samwell", "jon", "margaery", "daenerys", "tyrion" }; | replaceAll(people, 5, 'z', 'a')  Tests the result when none of the letter that needs to be replaced is present in any of the elements of the array | 0 |
| string people[ 5 ] = { "samwell", "jon", "margaery", "daenerys", "tyrion" }; | replaceAll(people, 5, 'a', 'z')  Tests the result when some elements of the array have the given letter to replace | 4 |
| string people[ 5 ] = { "samwell", "jon", "margaery", "daenerys", "tyrion" }; | replaceAll(people, -5, 'a', 'z')  Tests the result for when the inputted size of the array is negative | -1 |
| string data2[ 4 ] = { "happy", "days", "are here", "again" }; | replaceAll(data2, 3, 'a', 'z')  Tests for the result when the size inputted is lesser than the number of elements in the entire array | 3 |
| string data3[4] = {"happy", "happy", "days", "days"}; | removeDuplicatedValues(data3, 4)  Tests the result for when there are duplicated elements in the array | 2 |
| string data4[4] = {"happy", "days", "are here", "again"}; | removeDuplicatedValues(data4, 4)  Tests the result when each element in the array is unique | 0 |
| string data4[4] = {"happy", "days", "are here", "again"}; | removeDuplicatedValues(data4, -5)  Tests the result when the inputted size is lesser than or equal to zero | -1 |
| string data3[4] = {"happy", "happy", "days", "days"}; | removeDuplicatedValues(data3, 3)  Tests for the result when the size inputted is lesser than the number of elements in the entire array | 1 |
| string people[ 5 ] = { "samwell", "jon", "margaery", "daenerys", "tyrion" }; | shiftRight( people, 5, 3, "foo" )  Tests for the result when the amount to be shifted is lesser than the size of the array | 3 |
| string people[ 5 ] = { "samwell", "jon", "margaery", "daenerys", "tyrion" }; | shiftRight(people, 5, 25, "foo")  Tests the result when the amount to be shifted is greater than the size of the array | 5 |