## a.

In the standardizeRules function, I found it troublesome to delete rules from the three arrays. Also, I need to compare word1 and word2 in these rules in different orders (word1[i] = word1 [j] or wod1[i] = word2[j]). In the determineQuality function, the process of extracting words from the document is hard. I need to add ‘\0’ manually to each of the end of the Cstrings. Then, I need to use triple loops to compare words to the rules.

## b.

function standardizeRules:

if nRules is negative, return 0;

repeatedly until the last of the rules:

if current rule’s distance is not positive,

delete it and decrease the number of rules by 1

repeatedly until the end of the first word of the current rule:

convert the current character to lowercase

if the character is not a letter

delete the rule and decrease the number of rules by 1

repeatedly until the end of the second word of the current rule:

convert the current character to lowercase

if the character is not a letter

delete the rule and decrease the number of rules by 1

repeatedly until the rule before the current rule:

if there is a rule containing the same words as the current rule’s

compare these two rules distance and delete the one with shorter distance, or delete the first one if the distances are equal;

decrease the number of rules by 1

return the total number of rules;

function determineQuality:

repeatedly until the end of the document:

convert the current character to lower case

if it is a letter

add it to the current element of the word array

record the current word as not empty

if it is a space and the current word is not empty

add a terminal character to the current word

increase the length of the current word array

record the current word as empty

if the last character of the document is not a space, increase the length of the word array by 1

set all the rules as not used

repeatedly until the end of the word array

repeatedly until the last of the rules

if the current rule has not been used and the first word of the current rule is the same as the current word in the document

check within distance of the current word in the document:

if there is another word found to be equal to the second word of the current rule

set the current rule as used

increase quality by 1

return quality;

## c.

## test data for standardizeRules:

nRules=10

| distance | word1 | word2 | reason |
| --- | --- | --- | --- |
| 2 | mad | scientist | This case contains rules that have the same words but different distances, rules that have the same words and distances, rule with uppercase letters, rule with a non-letter character, rule with non-positive distance, and rule with an empty word. |
| 4 | deranged | robot |
| 1 | NEFARIOUS | PLOT |
| 3 | half-witted | assistant |
| 2 | robot | deranged |
| 1 | plot | Nefarious |
| 13 | have | mad |
| 2 | scientist | mad |
| -2 | im | mad |
| 2 |  | mad |

nRules = 0;

reason: nRules = 0;

nRules = -1;

reason: nRules is negative;

## test data for determineQuality:

nRules = 4

document = "The mad UCLA scientist unleashed a deranged evil giant robot."

| distance | word1 | word2 | reason |
| --- | --- | --- | --- |
| 2 | mad | scientist | General test. |
| 4 | deranged | robot |
| 1 | nefarious | plot |
| 13 | have | mad |

nRules = 4

document = "The mad UCLA scientist unleashed a deranged robot."

| distance | word1 | word2 | reason |
| --- | --- | --- | --- |
| 2 | mad | scientist | document with several spaces between two words |
| 4 | deranged | robot |
| 1 | nefarious | plot |
| 13 | have | mad |

nRules = 4

document = "\*\*\*\* 2014 \*\*\*\*"

| distance | word1 | word2 | reason |
| --- | --- | --- | --- |
| 2 | mad | scientist | document with no letters at all |
| 4 | deranged | robot |
| 1 | nefarious | plot |
| 13 | have | mad |

nRules = 4

document = " That plot: NEFARIOUS!"

| distance | word1 | word2 | reason |
| --- | --- | --- | --- |
| 2 | mad | scientist | document with spaces at the beginning, a non-letter character between words and uppercase letters. |
| 4 | deranged | robot |
| 1 | nefarious | plot |
| 13 | have | mad |

nRules = 4

document = "deranged deranged robot deranged robot robot"

| distance | word1 | word2 | reason |
| --- | --- | --- | --- |
| 2 | mad | scientist | document having several matches of the same rule |
| 4 | deranged | robot |
| 1 | nefarious | plot |
| 13 | have | mad |

nRules = 4

document = "Two mad scientists suffer from deranged-robot fever."

| distance | word1 | word2 | reason |
| --- | --- | --- | --- |
| 2 | mad | scientist | document having hyphen between two words, which should be treated as one word. There should be no match in this document. |
| 4 | deranged | robot |
| 1 | nefarious | plot |
| 13 | have | mad |

nRules = 4

document = "I'm upset that on Nov. 15th, 2014, my 2 brand-new BMW 750Lis were stolen!!”

| distance | word1 | word2 | reason |
| --- | --- | --- | --- |
| 2 | upset | that | document with a lot of non-letter characters between letters, which should be treated like “im upset that on nov th my brandnew bmw lis were stolen” with 3 matches. |
| 1 | nov | th |
| 2 | brandnew | lis |
| 4 | my | stolen |

nRules = 0

document = "I'm upset that on Nov. 15th, 2014, my 2 brand-new BMW 750Lis were stolen!!”

reason: no matching rules