1.obstacles:

a) how to delete bad rules

b) how to delete other characters

c) how to compare document with rules

2. standardizedRules

set non-positive number of rules to 0

repeatedly for each rule:

1) check if the rule has zero length

2)check if there is any non-alpha character in the rules

3)check if there is any repetition of rules

if there it is, exchange the less inclusive one to the current position

if either one of the three situations is true

exchange the rule in the current position with the last rule

decrement the number of rules

check the rule exchanged to current position

return the number of rules

determineQuality

repeatedly for each character in the document:

if the character is a letter

cast it to lower case letter and put it into a new C string

or if it is a space

put it into a new C string

repeatedly for each character in the new C string:

if the character is a letter put it in a new 2D array

if the character is a string following a letter,

start a new row

repeatedly for each rule:

repeatedly for each Cstring in the 2D array:

repeatedly for each positive integer distance lower than the distance assigned:

if a word in 2D array matches a word in the rule and a word in 2D array that positive integer distance behind the word former word matches the other word in the rule

increment the counter

return the counter after the biggest loop

3. right test:

number of rules= 4;

distance[4] = {

2, 4, 1, 13

};

word1 = {

"mad", "deranged", "nefarious", "have"

};

word2 = {

"scientist", "robot", "plot", "mad"

};

standardizedRules:

change the number of rules to -3 /check return value for non-positive number of rules

change word1[0] to “” /check the return value for empty array

change word1[0] to “mAd” /check if the rules are cast to lower case

change word1[0] to “m-d” /check non-alpha rule

change word1[0] to “m d” / spaces are not allowed in the rules

change word1[1] to “mad” and word2[1] to “have” /check the return value for repeated rules and if the larger distance is kept

determineQuality

number of rules= 4;

distance[4] = {

2, 4, 1, 13

};

word1 = {

"mad", "deranged", "nefarious", "have"

};

word2 = {

"scientist", "robot", "plot", "mad"

};

document="The mad UCLA scientist unleashed a deranged evil giant robot.” /normal case

document="The mad UCLA scientist unleashed a deranged robot.”/ consecutive spaces

document="\*\*\*\* 2014 \*\*\*\*”/no valid words and space

document=" That plot: NEFARIOUS!”/a sentence with capital letters and punctuations

document="deranged deranged robot deranged robot robot”//repeat match of the same rule