**Jacob Alspaw - HW2 Testing Report**

**isAlphabeticalOrder(String isAlphabeticalOrder)**

This method takes a string and checks each letter to see if the input string is in alphabetical order. If the string is in alphabetical order, then the return value will true. Else, the return value will be false.

> HW2.isAlphabeticalOrder("aZbc")

false

> HW2.isAlphabeticalOrder("ab!cZ")

true

> HW2.isAlphabeticalOrder("zabc")

false

> HW2.isAlphabeticalOrder("abcZ")

true

**removeNchars(String removeNcharsTest, int n, char removedCharacter)**

This method takes inputs string, char for the removed character, and int for the number of times the character will be removed. The method takes a string and removes the designated character “n” number of times. The result will be an output with the character removed “n” amount of times.

> HW2.removeNchars("Jacob Alspaw", 2, 'J')

"acob Alspaw"

> HW2.removeNchars("Jacob Alspaw", 2, 'a')

"Jcob Alspw"

> HW2.removeNchars("Jacob Alspaw", 2, 'w')

"Jacob Alspa"

> HW2.removeNchars("Jacob Alspaw", 1, ' ')

"JacobAlspaw"

> HW2.removeNchars("Jacob Alspaw", 0, 'a')

"Jacob Alspaw"

**moveXright(char movedLetter, moveXrightTest)**

This method takes a string and char as input and moves the first instance of the designated char one position to the right. The output is the string with the moved character. If the character is at the end of the string, then the position will not change.

> HW2.moveXright('J', "Jacob Alspaw")

"aJcob Alspaw"

> HW2.moveXright('a', "Jacob Alspaw")

"Jcaob Alspaw"

> HW2.moveXright('A', "Jacob Alspaw")

"Jacob lAspaw"

> HW2.moveXright('w', "Jacob Alspaw")

"Jacob Alspaw"

**bracketString(String originalString, String itemString)**

This method takes inputs of the original string and the item to be placed in brackets. If part of the original string is comparatively equal to the part to be put in brackets, then it is put in brackets. If there is overlap, the second instance of the item to be placed in brackets will be ignored. The output is the original string with brackets in the appropriate locations.

> HW2.bracketString("Alspaw", "Jacob")

"Alspaw"

> HW2.bracketString("Cabcabcabc", "cabc")

"Cab[cabc]abc"

> HW2.bracketString("Jacob Alspaw", " ")

"Jacob[ ]Alspaw"

> HW2.bracketString("cabcabcabc", "cabc")

"[cabc]ab[cabc]"

**moveAllXsRight(char movedLetter, String moveAllXsRightTest)**

This method takes a string and char as input and moves the all instances of the designated char one position to the right. The output is the string with the moved characters shifted to the right. If the character(s) is at the end of the original input string, then the position does not change.

> HW2.moveAllXsRight('X', "XJacob XAlspawX")

"JXacob AXlspawX"

> HW2.moveAllXsRight('X', "XXJacob XXAlspawXX")

"JXXacob AXXlspawXX"

> HW2.moveAllXsRight(' ', "Jacob Alspaw")

"JacobA lspaw"

> HW2.moveAllXsRight('X', "oXXXXXXXXXXXXXXo")

"ooXXXXXXXXXXXXXX"

**moveXdown(char x, String stringO)**

This method takes a char and a string as input. The first occurrence of the designated input character should be swapped with one other character of the string. The character the input character is swapped with should be after the next occurrence of the end-of-line character '\n', the same number of characters after the end-of-line character as the first occurrence is after either its previous end-of-line character or the start of the string, and before the next end-of-line character. If no such character exists in the string, the output should be the same as the input string.

> HW2.moveXdown('X', "XJacob\nAlexander\nAlspaw")

"AJacob

Xlexander

Alspaw"

> HW2.moveXdown('X', "JacXob\nAlexander\nAlspaw")

"Jacxob

AleXander

Alspaw"

> HW2.moveXdown('X', "JacobX\nAlexander\nAlspaw")

"Jacobn

AlexaXder

Alspaw"

> HW2.moveXdown('X', "Jacob\nXAlexander\nAlspaw")

"Jacob

AAlexander

Xlspaw"

> HW2.moveXdown('X', "Jacob\nAlexXander\nAlspaw")

"Jacob

Alexaander

AlspXw"

> HW2.moveXdown('X', "Jacob\nAlexanderX\nAlspaw")

"Jacob

AlexanderX

Alspaw"

> HW2.moveXdown('X', "Jacob\nAlexander\nXAlspaw")

"Jacob

Alexander

XAlspaw"

> HW2.moveXdown('X', "Jacob\nAlexander\nAlsXpaw")

"Jacob

Alexander

AlsXpaw"

> HW2.moveXdown('X', "Jacob\nAlexander\nAlspawX")

"Jacob

Alexander

AlspawX"

> HW2.moveXdown('X', "Jacob\nAlexander\nAlspawX\nis\nAWESOME")

"Jacob

Alexander

AlspawX

is

AWESOME"

> HW2.moveXdown('X', "JacXob")

"JacXob"

**moveXleft(char movedLetter, moveXleftTest)**

This method takes a string and char as input and moves the first instance of the designated char one position to the left. The output is the string with the moved character. If the character is at the beginning of the string, then the position will not change.

> HW2.moveXleft('a', "Jacob")

"aJcob"

> HW2.moveXleft('J', "Jacob")

“Jacob”

> HW2.moveXleft('Z', "Jacob")

"Jacob"

> HW2.moveXleft('b', "Jacob")

"Jacbo"

**moveXup(char x, String stringO)**

This method takes a char and a string as input. The first occurrence of the designated input character should be swapped with one other character of the string. The character the input character is swapped with should be after the next occurrence of the end-of-line character '\n', the same number of characters after the end-of-line character as the first occurrence is after either its previous end-of-line character or the start of the string, and before the next end-of-line character. If no such character exists in the string, the output should be the same as the input string.

> HW2.moveXup('X', "Jacob\nJacob")

"Jacob

Jacob"

> HW2.moveXup('X', "Jacob\n\nJacobX")

"Jacob

JacobX"

> HW2.moveXup('X', "Jacob\nJacob\nJaXcob")

"Jacob

JaXob

Jaccob"

> HW2.moveXup('X', "Jacob\nJacobX\nJacob")

"Jacob

JacobX

Jacob"

> HW2.moveXup('X', "Jacob\nXJacob\nJacob")

"Xacob

JJacob

Jacob"

> HW2.moveXup('X', "JacXob\nJacob\nJacob")

"JacXob

Jacob

Jacob"

> HW2.moveXup('X', "Jacob\nJacob\nJacob\nJacoXb")

"Jacob

Jacob

JacoX

Jacobb"

> HW2.moveXup('X', "Jacob\nJacob\nJacobX")

"Jacob

Jacob

JacobX"

> HW2.moveXup('X', "Jacob\nJacob\nXJacob")

"Jacob

Xacob

JJacob"

> HW2.moveXup('X', "Jacob\nJacXob\nJacob")

"JacXb

Jacoob

Jacob"

> HW2.moveXup('X', "JacobX\nJacob\nJacob")

"JacobX

Jacob

Jacob"

> HW2.moveXup('X', "XJacob\nJacob\nJacob")

"XJacob

Jacob

Jacob"

> HW2.moveXup('X', "Jacob\nOOX\nJacob")

"JaXob

OOc

Jacob"