Overview

The meanings of the different test cases will be explained for each method tested, then the reasons for the choosing of those test cases will be addressed.

replaceFirstK

* Tests zero, one, and many signify that the integer input is zero, one, or more than one respectively. This is covered because it is important that the method can work with any limit
* Tests first, middle, and last signify that the character that is being replaced comes first, in the middle, or last in the string respectively. This is covered because it is important that the method can replace a character regardless of its position
* Tests limited and underlimited signify that the replace limit is either less than or greater than the occurrences of the character it is replacing, respectively. This is covered because it is important that the method abides by the replace limit and that the method does not break when it is underlimited

allChars

* Tests zero, one, and many signify that the inputs are out of order (and thus should have zero chars between them), are identical (and thus should have no chars between them other than the char that they are set to), or are far apart in correct order (and thus should have other chars between them) respectively. This is covered because it is important that the method can work in all cases, even if the inputs are strange

showCharOfString

* Tests zero, one, and many signify that the exceptions string (b in the code) has zero, one, or more than one characters in it respectively. This is covered because it is important that the method can handle replacing everything, all but some, or no characters according to the exceptions string
* Tests first, middle, and last signify that the character to be kept comes first, in the middle, or last in the string respectively. This is covered because it is important that the method can keep a character regardless of its position
* Test zero body string signifies that the body string (a in the code) has zero characters in it. This is covered because it is important that the program does not append any extraneous chars

hiddenInString

* Tests zero, one, and many signify that the integer k is equal to zero, one, or more than one respectively. This is covered because it is important that the code searches for strings with characters spaced k apart correctly
* Test negative signifies that the integer k is negative. This is covered because it is important that the code can search for a string backwards, spaced k apart.
* Tests first, middle, and last signify that the string to find occurs in the searched string at the beginning, in the middle, or end respectively. This is covered because it is important that the code can find a string regardless of its position
* Tests zero search string and zero body string signify that the search string (searchFor in the code) or the body string has zero characters in it, repsectively. This is covered because it is important that the method can handle strange inputs
* Test search string not in body spaced by k signifies that the correctly spaced search string is not in the body string. This is covered because it allows for testing whether or not the code is searching by spacing k as long as the search string is actually in the body string, just not spaced by k

capitalizeWords

* Tests zero, one, and many signify that the input string contains zero, one, or more than one word, respectively. This is important because the method should be able to handle strings of any length
* Tests first, middle, and last signify that the word with a capital letter in it comes at the beginning, in the middle, or at the end, respectively. This is covered because it is important that the code can capitalize a word regardless of its position
* Test capital signifies that the input contains a capital letter. This is covered because it is important that the code can capitalize the only word in a one-word string