

#1 Test

The purpose of this method is to check whether or not the string is full of letters. It returns false if any of the characters are not letters, it returns false. I first tested a string of letters followed by a string of letters and numbers.

Test Code:

```
System.out.println(HW2.onlyEnglishLetters("abDkfdofRs"));
System.out.println(HW2.onlyEnglishLetters("lFKe5aaa"));
```

Output:

```
true
false
```

#2 Test

The purpose of this method is alter the n-th appearance of a character 'a' and replace it with 'b'. I first tested the case where the string length was larger than the number passed to it. Then the test was of a string length smaller than the number. Lastly, if the number was 0 (which none should be replaced).

Test Code:

```
System.out.println(HW2.replaceKth('a', 'x', 3, "abcaaa"));
System.out.println(HW2.replaceKth('a', 'x', 6, "aaaaa"));
System.out.println(HW2.replaceKth('a', 'x', 0, "aaaaa"));
```

Output

```
abcaxa
aaaaa
aaaaa
```

#3 Test

The purpose of this method is the interweave the first string and last string such that the strings alternate and any extra characters of a longer string are just added to the end. I checked each case where one string was longer than the other and if any of the strings were empty.

Test Code:

```
System.out.println(HW2.interleave("abcde", "ABC"));
System.out.println(HW2.interleave("ABC", "abcde"));
System.out.println(HW2.interleave("abcde", ""));
System.out.println(HW2.interleave("", "abcde"));
```

Output:

```
aAbBcCde
AaBbCcde
abcde
abcde
```

#4 Test

The purpose of this method is to blank out the middle characters of a word as long as the word is a minimum of three characters. I tested the case where there were words longer than three characters and where there wasn't. I also tested an Empty String.

Test Code:

```
System.out.println(HW2.blankWords("This is a Test."));
System.out.println(HW2.blankWords("Hi."));
System.out.println(HW2.blankWords(""));
```

Output:

```
T__s is a T__t.
Hi.
(this is the whitespace output)
```

#5 Test

The purpose of this method is to only output the first and every n-th word of a given string. I tested the case where the number was less than the number of words in the area, more than, and the case where there was only one string in the array.

Test Code:

```
System.out.println(HW2.nthWord(3, "zero one  two  three four five six seven"));
System.out.println(HW2.nthWord(3, "zero"));
System.out.println(HW2.nthWord(9, "zero one  two  three four five six seven"));
```

Output:

```
zero three six
zero
zero
```

#6 Test

The purpose of this method is to output the min number of letters truncated to a safe word length. I tested the case where we are in the middle and at the end of a word.

Test Code:

```
System.out.println(HW2.truncateAfter(6, "La-te-ly the-re."));
System.out.println(HW2.truncateAfter(7, "La-te-ly the-re."));
```

Output:

Lately
Lately the-

EC Test

The purpose of this method is to output the max number of letters truncated to a safe location. I tested the cases where it was at the beginning, middle, end and middle of the first word cases.

Test Code:

```
System.out.println(HW2.truncateBefore(4, "La-te-ly the-re."));  
System.out.println(HW2.truncateBefore(5, "La-te-ly the-re."));  
System.out.println(HW2.truncateBefore(6, "La-te-ly the-re."));  
System.out.println(HW2.truncateBefore(7, "La-te-ly the-re."));
```

Output:

Late-
Late-
Lately
Lately