Strings Practice

10-11 are from previous worksheets and slides, consolidated here for convenience. I recommend doing the shorter, more mechanical problems first to get warmed up.

Problems 2-8 are adapted from Stanford's Nick Parlante.

Quick ones to do first:

1. Write a function that prompts the user for a word and prints out its equivalent in Pig Latin. To translate a word to Pig Latin, take the initial letter, move it to the end of the word and add 'ay'. The new suffix (first letter with ay) should be printed in all caps. Here is a sample run of the program:

   ```
   Give me a word to translate to Pig Latin: trash
   Your word is ashtRAY in Pig Latin!
   ```

   Challenge: can you translate a full sentence from the user? Separating words will take some cleverness.

4. Write a method `countExclaim` that given a string, returns how many '!' characters it contains. For example, `countExclaim("Ow! My thumb! Stop hammering it, Kate!")` should return 3.

5. Write a method `countChar` that takes two parameters: a string and a character. It should be a generalization of the previous method such that `countChar("Hello, world!", 'o')` should return 2. Notice the single quotes around 'o.'

6. Write a method that given a string returns a string made of repetitions of that string. For example, a call on `repeat("Tacos", 5)` should return "TacosTacosTacosTacosTacos"

7. Write a method that given a string with at least two dashes returns the text between the first two. For example, the call `middleText("You may -absolutely- not go to the bathroom")` should return "absolutely."

8. Given a string, return true if the first 2 chars in the string also appear at the end of the string, such as with "edited".

   ```
   frontAgain("edited") → true
   frontAgain("jello") → false
   frontAgain("ed") → true
   ```

9. A sandwich is two pieces of bread with something in between. Return the string that is between the first and last appearance of "bread" in the given string, or return the empty string "" if there are not two pieces of bread.

   getSandwich("breadjambread") → "jam"
   getSandwich("xxbreadjambreadyy") → "jam"
   getSandwich("xxbreadyy") → ""

More interesting/useful ones:

10. Write a method named isAllVowels that returns whether a String consists entirely of vowels (a, e, i, o, or u, case-insensitively). If every character of the String is a vowel, your method should return true. If any character of the String is a non-vowel, your method should return false. Your method should return true if passed the empty string, since it does not contain any non-vowel characters.

    For example, here are some calls to your method and their expected results:

Call Value	Returned
`isAllVowels("eIEiO")`	`true`
`isAllVowels("oink")`	`false`

11. A Caesar cipher is a simple encryption scheme in which a message is encoded by shifting each letter by a given amount. For example, with a shift of 3, A goes to D, H goes to K, X goes to A and Z goes to C.

    Write a method that reads a message from the user and performs a Caesar cipher on its letters as follows:

```
Your secret message: Brad thinks Angelina is cute
Your secret key: 3
The encoded message: eudg wklqnv dqjholqd lv fxwh
```

Impress me:

12. Given a string, does "xyz" appear in the middle of the string? To define middle, we'll say that the number of chars to the left and right of the "xyz" must differ by at most one. This problem is harder than it looks.

    xyzMiddle("AAxyzBB") → true
    xyzMiddle("AxyzBB") → true
    xyzMiddle("AxyzBBB") → false

13. Given a string, return the sum of the numbers appearing in the string, ignoring all other characters. A number is a series of 1 or more digit chars in a row. (Note: Character.isDigit(char) tests if a char is one of the chars '0', '1', .. '9'. Integer.parseInt(string) converts a string to an int.)

    sumNumbers("abc123xyz") → 123
    sumNumbers("aa11b33") → 44
    sumNumbers("7 11") → 18