

Methods in Computational Linguistics I

Practicum 3

Practice assignments

1. Write a function that takes a string X and returns a string consisting of the first and last characters of X.
2. Write a function that takes a list of words X, and returns a list Y whose items are what's returned after passing each X's words to the function from the previous problem.
3. Write a function that takes a string of words and returns a string of words, but each word in the returned string is converted to only its first and last letters (you may use the function(s) from the above problem
 - a. Example: "this just in" → "ts jt in"
4. Write a function that takes a string of words and output the number of vowels
 - a. Try to factor in the fact that y isn't always considered a vowel
5. Write a function that takes a list of words, and outputs a dictionary whose keys are the words in the list and whose values are the number of times the word is repeated in the list.
6. Write a function that prints the contents of a string in the following "crazy" way:
 - a. The crazy version of a string X consists of all and only the characters of X, but rearranged in this order: first character, last character, second character, second to last character, etc... the input and output strings should be the same length. The program should work if the input string has an even or odd number of characters (note that is theres an odd number, the middle letter should be printed as well, and only once). Examples
 - i. "a" → "a"
 - ii. "ab" → "ab"
 - iii. "abc" → "acb"
 - iv. "abcd" → "adbc"
 - v. "abcde" → "aebdc"

Note: It may help to work out some math or patterns on paper if using for loops and/or the indices of the items in a list. One approach might be more math-oriented (in terms of the indices of the characters in the string), while another type of approach might be less so and might involve starting with a initial string, appending parts of it to a new string, and replacing the initial strings value with some substring of itself, and repeating this process in a loop.

7. Pig latin translation
 - a. Make a program that takes a string, and employs rules of pig latin to return a string that conforms as closely as possible. The more closely you attempt to adhere to these rules, and depending on the rules you use (note that the real rules may be based on phonology and not orthography!) a perfect translator will not be realistic. One way to do this is to start with something very basic, but also very rough, and tweak your code from there. Feel free to create multiple methods, or just one more complex method.

