Start

is_vowel -> character to bool, is_vowel returns true when its argument is a vowel and false otherwise.

is_consonant -> character to bool, is_consonant returns true when its argument is a consonant and false otherwise.

ends_with -> two strings, candidate and suffix (in that order) to bool
If candidate is an empty string and suffix is an empty string, ends_with returns true. If candidate is an empty string and suffix is not an empty string, ends_with returns false. If candidate ends with suffix, ends_with returns true. Otherwise, it returns false.

ends_with_double_consonant a string to bool
ends_with_double_consonant returns true if the last two characters of the argument are both consonants, and equal to one another. It returns false otherwise.

ends_with_cvc string to bool,
ends_with_cvc returns true if the last three characters of the argument are a consonant, a vowel and then a consonant (hence the name, cvc). It returns false otherwise.

count_consonants_at_front -> string to int
count_consonants_at_front returns the number of consecutive constants at the beginning of the argument. For example, count_consonants_at_front (std::string{"threw"}) is 3.

count_vowels_at_back string to int
count_vowels_at_back returns the number of consecutive vowels at the end of the argument. For example, count_vowels_at_back(std::string{"free"}) is 2. The string passed as an argument may be empty.

contains_vowel -> string to bool
contains_vowel returns true if there is a vowel anywhere in the argument. It returns false otherwise. The string passed as an argument may be empty.

new_ending -> string, a number and a string named candidate, suffix length and replacement, respectively (and in that order) to string
new_ending creates a new string from candidate by removing its last suffix length characters and replacing them with replacement. new_ending returns that new string. For example, new_ending(std::string{"testing"}, 3, std::string{"ed"}) is "tested". You may assume that suffix length is always less than the length of candidate.