- Define a function is_palindrome() that recognizes palindromes (words that look the same backwards).
  - For example, is_palindrome("radar") should return True.
  - Try to use only one index in your loop.

- Write a program which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200 (both included). The numbers obtained should be printed in a comma-separated sequence on a single line (hint: use the join function).
- Write a function encrypt() that gets a string and returns a translated string by doubling every **consonant** (and not a vowel). E.g, "this is so fun" should be translated to "tthhiss iss sso ffunn".
- Write the opposite function decrpyt()

Exercises (3)

- Write a recursive function that takes a number and prints the “99 bottles” song with the following format:
  - 99 bottles of beer on the wall, 99 bottles of beer.
    Take one down, pass it around, 98 bottles of beer on the wall.
- Write a function that gets a list of strings and returns the longest string. The list can also contain inner lists of strings, so you need to look inside those lists as well.
  - Hint: Use the *isinstance*() function.

List Comprehensions Exercises

- Write a single-line command that maps a list of words into a list of integers representing the lengths of the corresponding words.
- Now take another list of words and create a list by concatenating each element in the first list with each element in the second list.
- Write a single-line function *twoLetterSubs* that gets a list of words and returns a list of all two-letter substrings that appear in those words.
  - For example: twoLetterSubs(['hello','world']) should return ['he', 'el', 'll', 'lo', 'wo', 'or', 'rl', 'ld'].