Tutorial Objectives

- Practice working with **recursion**

**Problem 1**

Write a *recursive* function called `pow(b,x)` that takes two integers as arguments: a base value `b`, and an exponent `x`, and returns the result of `b^x`. Note, your solution should not use the `**` operator nor Math.exp(). Instead, implement the exponent process using repeated multiplication. E.g.: $3^4 = 3*3*3*3$.

**Problem 2**

Write a function called `palindrome(s)` that takes a string as argument and returns true if that string is a palindrome. A palindrome is a word or phrase that spells the same thing (ignoring spaces) forwards as backwards. Your program should use a recursive process to determine the result.

**Problem 3**

Write a function called `filterVowels(s)` that takes a string as argument and returns a string with all of the vowels removed. Again, your function should make use of a recursive process, not the `str.replace()` function.

E.g. `filterVowels("Hello World!")` → "Hll Wrld!

**Problem 4**

Write a recursive function called `squareList()` that takes a list of numbers as argument and returns a list with every value in the original squared.

E.g. `squareList([1,2,3,4,5])` → [1, 4, 9, 16, 25]

**Problem 5**

Write a function called `maxList(L)` that takes a list as argument and (using a recursive process) returns the largest item in the list.

E.g. `maxList( [5,7,4,6,4,2,8,3] )` → 8
Extra practice, Problem 6

Write a function count(s, token) that takes two strings as argument, a source, and a token. The function should use a recursive process to count the number of times the token appears in the given string. For simplicity, you may assume the token is only a single character long. For a greater challenge, allow for any sized token.

E.g. count("Hello World!", "o") → 2

Extra practice, Problem 7

Write a function called f(n) that computes the nth number in the series defined by the following function:

\[ f(x) = \begin{cases} 
  x, & \text{if } x < 3 \\
  3 \cdot f(x - 1) + 2 \cdot f(x - 2) + f(x - 3), & \text{otherwise}
\end{cases} \]

E.g.:
- f(1) → 1
- f(7) → 1382
- f(25) → 16336148470525

For an extra challenge, try expressing this function using iteration (i.e., using a for or while loop).