Problem 1.

Recall that in lecture, we wrote the factorial function as follows:

```
def fact(n):
  if n == 0:
    return 1              # base case
  else:
    return n * fact(n-1)  # recursive step
```

We constructed the *call tree* for `fact(3)` as follows:

```
fact(0)
      \
      /1
fact(1)
      \
      /1 * 1
fact(2)
       \
       /2 * 1 * 1
fact(3)
       \
       \3 * 2 * 1 * 1
```

Here, we write a function that computes `x^n` recursively, using only multiplication.

Part (a)

How do you decompose x^n into smaller problems, in the same way we decomposed $fact(n) = n \times fact(n-1)$?
What is the base case, if n is non-negative?
What is the recursive step?

Part (b)

Write the function `power(x, n)` that computes x^n recursively, using only multiplication.

Part (c)

Construct the call tree for `power(2, 3)`.

Problem 2.

Write a function that sums up the digits of a number. For example, the sum of the digits of 123 is 1+2+3 = 6. The function must be recursive.
Hint: what is the base case? What is the recursive step?

Problem 3.

This question is not meant to be about recursion.

In Project 3, you need to split up a string into sentences, with the sentences being separated by punctuation marks.

Here, you will do a similar problem with lists.

Write a function that splits up a list into sublists, where the sublist are elements of the original list separated by a given list of elements. For example, if the list is `[1, 2, 6, 4, 5, 3, 7]` and the list of elements is `[3, 6]`, the function should return `[[1, 2], [4, 5], [7]]`.

Hint: write a helper function that splits up a list into sublists separated by a single element. Then, use this helper function to split up the list by multiple elements. You can first replace e.g., every separator by the first element in the separator list, and then split up the list by that element.

Note: for strings, you can and should use the functions `str.split()` and `str.replace()`. However, for lists, you need to write your own function.

Problem 4.

Work on Project 3.

Problem 5.

This is the last lab, say bye to the TAs!