



Bookmarks



Bookmark

► Overview

► Entrance
Survey

► Week 1

► Week 2

▼ Week 3

**Lecture 5 -
Recursion - Time
55:06**

Lecture Sequence

**Lecture 6 - Objects
- Time 46:20**

Lecture Sequence

**Complete
Programming
Experience: ndigits**

Problem Set 3

Problem Set due Jul 04,
2016 at 23:30 UTC

► Week 4

► Sandbox

Week 3 > Lecture 5 - Recursion - Time 55:06 > L5 Problem 9

L5 Problem 9

(5 points possible)

ESTIMATED TIME TO COMPLETE: 20 minutes

A **semordnilap** is a word or a phrase that spells a different word when backwards ("semordnilap" is a semordnilap of "palindromes"). Here are some examples:

- nametag / gateman
- dog / god
- live / evil
- desserts / stressed

Write a recursive program, `semordnilap`, that takes in two words and says if they are semordnilap.

This recursive function is not entirely straightforward. There are a few things that you need to check the first time you look at the inputs that you should not check on subsequent recursive calls: you need to make sure that the strings are not single characters, and also you need to be sure that the strings are not equal. If you do this check every time you call your function, though, this will end up interfering with the recursive base case (which we don't want!).

There's a few different ways you can perform checks on the inputs the first time. The first way would be to use *keyword arguments*. The second way would be to use a *global variable*, which you'll see in the next lecture video; however, using global variables is always a bit risky and thus not the best way to do this.

The third way to perform checks on the inputs the first time you see them, but not any subsequent time, is to use a *wrapper function*. This wrapper function performs some checks, then makes a call to the recursive function.



The idea of a wrapper function is really important. You'll see more wrapper functions later. To introduce you to the idea, we are providing you with the wrapper function; your job is to write the recursive function `semordnilap` that the wrapper function calls. Here is the wrapper function:

```
def semordnilapWrapper(str1, str2):
    # A single-length string cannot be semordnilap
    if len(str1) == 1 or len(str2) == 1:
        return False

    # Equal strings cannot be semordnilap
    if str1 == str2:
        return False

    return semordnilap(str1, str2)
```

Fill in the definition for `semordnilap` in the box below.

```
1 def semordnilap(str1, str2):
2     ""
3     str1: a string
4     str2: a string
5
6     returns: True if str1 and str2 are semordnilap;
7             False otherwise.
8     ""
9     # Your code here
10
```

Unanswered

Note: In programming there are many ways to solve a problem. For your code to check correctly here, though, you must write your recursive function such that you make a recursive call directly to the function `semordnilap`. Thank you for understanding.

Hints

Unequal Strings

What happens when two strings are not of equal length? What can you immediately return?

What should your base case be?

What happens when the strings are of length 1?

What should your recursive case be?

Go back to the definition of a semordnilap. What can you say about the first letter of `str1` and the last letter of `str2`?

If you are getting the error stating that "Your code should be recursive" when you already make a call to `semordnilap` : check your indention -- specifically, a common mistake is that your function and docstring do not start at the same indentation level.

© All Rights Reserved



© edX Inc. All rights reserved except where noted. EdX, Open edX and the edX and Open EdX logos are registered trademarks or trademarks of edX Inc.

