



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 &gt; Lecture 5 - Recursion - Time 55:06 &gt; L5 Problem 8

## L5 Problem 8

(5 points possible)

**ESTIMATED TIME TO COMPLETE: 18 minutes**

We can use the idea of **bisection search** to determine if a character is in a string, so long as the string is sorted in alphabetical order.

First, test the middle character of a string against the character you're looking for (the "test character"). If they are the same, we are done - we've found the character we're looking for!

If they're not the same, check if the test character is "smaller" than the middle character. If so, we need only consider the lower half of the string; otherwise, we only consider the upper half of the string. (Note that you can compare characters using Python's `<` function.)

Implement the function `isIn(char, aStr)` which implements the above idea recursively to test if `char` is in `aStr`. `char` will be a single character and `aStr` will be a string that is in alphabetical order. The function should return a boolean value.

As you design the function, think very carefully about what the base cases should be.



## 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 `isIn`. Thank you for understanding.

### Hints

Basic function structuring

Be very careful about how you slice the string in the recursive cases! Before you execute the recursive cases, you test the middle character - so if you reach the recursive cases, you know the middle character cannot be a match, right? So be careful to *not* include this character when you make your recursive call!

What should your base case be?

You should be thinking about 3 situations:

- What happens when the string is empty?
- What happens when the string is of length 1?
- What happens when the test character equals the middle character?

What should your recursive case be?

You should be thinking about 2 situations:

- What happens when the test character is smaller than the middle character?
- What happens when it is larger?

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



© 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.

