(/) **MITx: 6.00x Introduction to Computer Science and Programming**

Ankit_Aggarwal (/dashboard) ▾

Courseware (/courses/MITx/6.00x/2012_Fall/courseware) **Course Info (/courses/MITx/6.00x/2012_Fall/info)**

Textbook (/courses/MITx/6.00x/2012_Fall/book/0/) **Discussion (/courses/MITx/6.00x/2012_Fall/discussion/forum)**

Wiki (/courses/MITx/6.00x/2012_Fall/course_wiki) **Progress (/courses/MITx/6.00x/2012_Fall/progress)**

PROBLEM 4: ERICIAN : 15.0 POINTS

A word is considered *erician* if it contains the letters `e`, `r`, `i`, and `c` in it, in that order. For example, we would say that the following words are *erician*: "meritocracy", "generic", "derrick", "euphoric", "heretic", and "electric", because they each contain those four letters in the correct order. The word "rice" is *not* erician because the four letters appear in the wrong order.

In this problem, we want you to write a more generalized function called `x_ian(x, word)` that returns `True` if all the letters of `x` are contained in `word` in the same order as they appear in `x`. For example:

```
>>> x_ian('eric', 'algebraic')
True
>>> x_ian('john', 'mahjong')
False
>>> x_ian('alvin', 'palavering')
True
>>> x_ian('sarina', 'czarina')
False
```

This function has to be recursive! You may not use loops (`for` or `while`) to solve this problem.

```
def x_ian(x, word):
    """
    Given a string x, returns True if all the letters in x are
    contained in word in the same order as they appear in x.

    x: a string
    word: a string
    returns: True if word is x_ian, False otherwise
    """
    ###TODO

```

Unsubmitted

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 `x_ian`. Thank you for understanding.

Check **Save** *You have used 0 of 30 submissions*

Show Discussion **New Post**