

[Courseware \(/courses/MITx/6.00x/2012_Fall/courseware/\)](/courses/MITx/6.00x/2012_Fall/courseware/)[Course Info \(/courses/MITx/6.00x/2012_Fall/info/\)](/courses/MITx/6.00x/2012_Fall/info/)[Textbook \(/courses/MITx/6.00x/2012_Fall/book/0/\)](/courses/MITx/6.00x/2012_Fall/book/0/)[Discussion \(/courses/MITx/6.00x/2012_Fall/discussion/forum/\)](/courses/MITx/6.00x/2012_Fall/discussion/forum/)[Wiki \(/courses/MITx/6.00x/2012_Fall/course_wiki/\)](/courses/MITx/6.00x/2012_Fall/course_wiki/)[Progress \(/courses/MITx/6.00x/2012_Fall/progress/\)](/courses/MITx/6.00x/2012_Fall/progress/)

PROBLEM 5: TYPEWRITER : 15.0 POINTS

Imagine a typewriter: whenever it's in the middle of a word, and reaches its desired line length, the internal bell rings. This signifies to the typist that *after* finishing the current word, a newline must be manually inserted. Check out this cool video (<http://www.youtube.com/watch?v=g2LJ1i7222c>) to see what it looks like! We ask you to emulate this process such that a newline is inserted as required after each word that exceeds the desired line length. Note that if the typewriter's bell rings on a space, a newline should be inserted before the start of the next word.

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

Hints

A Few Helpful Thoughts

- Write helper functions as appropriate. If you wish to use `insertNewlines` as a wrapper function that makes an appropriate call to a recursive function, please name your recursive helper function `insertNewlinesRec` so it can be properly graded by our automatic grader.
- `lineLength` is *not* the maximum number of characters in the line. It is the length *after* which the next word should be wrapped to the next line.
- Make sure that if a space occurs on the index of the desired line length, the next word is wrapped to the next line.

Test Cases

Test case 1:

```
>>> print insertNewlines('While I expect new intellectual adventures ahead, nothing will compare to the exhilaration of the world-changing accomplishments that we produced together.', 15)
```

```
While I expect
new intellectual
adventures ahead,
nothing will compare
to the exhilaration
of the world-changing
accomplishments
that we produced
together.
```

Test case 2:

```
>>> print insertNewlines('Nuh-uh! We let users vote on comments and display them by number of votes. Everyone knows that makes it impossible for a few persistent voices to dominate the discussion.', 20)
```

Nuh-uh! We let users vote on comments and display them by number of votes. Everyone knows that makes it impossible for a few persistent voices to dominate the discussion.

```
1 def insertNewlines(text, lineLength):
2     """
3     Given text and a desired line length, wrap the text as a typewriter would.
4     Insert a newline character ("\n") after each word that reaches or exceeds
5     the desired line length.
6
7     text: a string containing the text to wrap.
8     lineLength: the number of characters to include on a line before wrapping
9                 the next word.
10    returns: a string, with newline characters inserted appropriately.
11    """
12    ### TODO
13
```

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 *either* the function `insertNewlines` or - if you wish to use a helper function - `insertNewlinesRec`. Thank you for understanding.

Check

Save

You have used 0 of 30 submissions

Show Discussion

New Post



[Find Courses \(/courses\)](/courses) [About \(/about\)](/about) [Blog \(http://blog.edx.org/\)](http://blog.edx.org/) [Jobs \(/jobs\)](/jobs) [Contact \(/contact\)](/contact)



(<http://youtube.com/user/edxonline>)



(<https://plus.google.com/108235383044095082735>)



(<http://www.facebook.com/EdxOnline>)



(<https://twitter.com/edXOnline>)

© 2012 edX, some rights reserved.

[terms of service \(/tos\)](/tos)

[privacy policy \(/privacy\)](/privacy)

[honor code \(/honor\)](/honor)

[help \(/help\)](/help)