HW 1-3: ANAGRAMS

An anagram is a word obtained by rearranging the letters of another word. For example, "rats", "tars", and "star" are anagrams of one another, as are "dictionary" and "indicatory". We will call any array of single-word anagrams an anagram group. For instance, `["rats", "tars", "star"]` is an anagram group, as is `["dictionary"]`.

Write a method `combine_anagrams(words)` that, given an array of strings `words`, groups the input words into anagram groups. Case doesn't matter in classifying strings as anagrams (but case should be preserved in the output), and the order of the anagrams in the groups doesn't matter. The output should be an array of anagram groups (i.e. an array of arrays).

Code skeleton:

```
def combine_anagrams(words)
    # your code here
end
```

Example test case:

```
# input: ['cars', 'for', 'potatoes', 'racs', 'four', 'scar', 'creams', 'scream']
# output: [ ["cars", "racs", "scar"],
#           ["four"],
#           ["for"],
#           ["potatoes"],
#           ["creams", "scream"] ]
```

Hint: You can quickly tell if two words are anagrams by sorting their letters, keeping in mind that upper vs. lowercase doesn't matter.

(Choose Files) No file chosen

Check

Show Discussion   **New Post**

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

(http://youtube.com/user/edxonline)   (https://plus.google.com/108235383044095082735)

(http://www.facebook.com/EdxOnline)   (https://twitter.com/edXOnline)

terms of service (/tos)   *privacy policy (/privacy)*   *honor code (/honor)*   *help (/help)*