Programming for Computational Linguistics 2025/2026

Exercise Set 5 (2025-11-20)
Strings

Once you have completed all exercises, send an email to the progclgrader@ims.uni-stuttgart.de with the title "submit set5", and with your python files as attachments. You should receive a response with feedback and a grade.

Exercises are due on **2025-12-09 23:59**, but we encourage you to do them during the lab session.

Exercise 1. In a file called `rot.py`, write a function `rotated(lst, n)`. The function should take a list `lst` and an int `n` as input, and should return a new list that is a copy of `lst` "rotated" n positions to the right. For example, `rotated([1, 2, 3, 4, 5], 1)` should return `[2, 3, 4, 5, 1]`, and `rotated([1, 2, 3, 4, 5], 3)` should return `[4, 5, 1, 2, 3]`. If n is negative, `lst` should be rotated to the left instead of the right. Some example input-output pairs are given below to clarify the expected behavior:

```
>>>rotated([1, 2, 3], 0)
[1, 2, 3]
>>>rotated([1, 2, 3], 1)
[2, 3, 1]
>>>rotated([1, 2, 3], 2)
[3, 1, 2]
>>>rotated([1, 2, 3], 3)
[1, 2, 3]
>>>rotated([1, 2, 3], 4)
[2, 3, 1]
>>>rotated([1, 2, 3], −1)
[3, 1, 2]
>>>rotated([1, 2, 3], −2)
[2, 3, 1]
>>>rotated([1, 2, 3], −3)
[1, 2, 3]
>>>rotated([1, 2, 3], −4)
[3, 1, 2]
```

Make sure that your function does not modify `lst` – it should return a new list.

Exercise 2. You are a restaurant owner plagued by bad online reviews. You want to write a function to turn these bad reviews into good reviews. In a file called `positivity.py`, write a function `positivize(review)` that takes review text `review` as a string. Your function should return a "positivized" version of the review, where:

- Every instance of the word "bad" should be replaced by "good"
- Every instance of the word "horrible" should be replaced by "fantastic"
- Every instance of the word "dirty" should be replaced by "clean"
- Every instance of the word "disgusting" should be replaced by "sublime"
- Every instance of the word "expensive" should be replaced by "affordable"
- Every instance of the word "moldy" should be replaced by "flavourful"
- Every instance of the word "frozen" should be replaced by "farm-fresh"
- Every instance of the phrase "n minutes" should be replaced by "only $n/2$ minutes"

Your function should preserve the original review's capitalization – i.e. "Dirty" should map to "Clean." You can assume the original review is capitalized sensibly.
Example: The string

> `The food was horrible!!!  We waited 40 minutes for frozen vegetables and moldy bread. Disgusting!`

should map to

> `The food was fantastic!!!  We waited only 20 minutes for farm-fresh vegetables and flavourful bread.  Sublime!`

Exercise 3. In a file called `order.py`, write a function `sort(words)`, which takes as an argument a list of strings `words`. `sort` should modify `words` so as to sort the elements in alphabetical order. Python provides some built-in functions to do this, such as `list.sort` and `sorted` – **do not use these for this exercise**. Your solution should only use "basic" components for its logic, such as list access, list assignments, variables, if-statements, loops. In particular, **the only built-in function you may use for this exercise is `len`**.

Do not worry about efficiency, so long as your program can sort short lists in a reasonable amount of time.
Example:

```
>>>animals = ['cat', 'bat', 'zebra', 'fish', 'dog']
>>>sort(animals)
>>>animals
['bat', 'cat', 'dog', 'fish', 'zebra']
```

Hint: Read about bubble sort