Assignment #6: Strings

Master in Informatics and Computing Engineering
Programming Fundamentals
Instance: 2018/2019

0. Introduction

Goals*: To write functions using strings and string operations*

Pre-requirements (prior knowledge)*: See bibliography of Lecture #10*

Rules*: You may work with colleagues, however, each student must write and submit in Moodle his or her this assignment separately. Be sure to indicate with whom you have worked. We may run tools to detect plagiarism (e.g.duplicate code submitted)*

Deadline*: 8:00 Monday of the week after (12/11/2018)*

Submission*: to submit, first pack your files in a folder RE06, then compress it with zip to a file with name 2018xxxxx.zip (your_code.zip) and last (before the deadline) go to the Moodle activity (**you have only 2 attempts**)*

1. String iterator

Write a Python function `rm_letter_rev(l, astr)`t that removes all occurrences of a given letter `l` (case sensitive) from the given string `astr` (case sensitive), reverses the order of the characters and return the resulting string.

Save the program in the file `rm_letter_rev.py`

For example:

- for `l="s"` and `astr="A Style Guide is about consistency"`, the function returns the string `"ycnetinoc tuoba i ediuG elytS A"` (without quotes)
- for `l=" "` and `astr="a nut for a jar of tuna"`, the function returns the string `"anutforajaroftuna"` (without quotes)

2. Count subset

Write a Python function `count(word, phrase)` which counts how many times `word` appears in the `phrase`. Upper and lower cases are to be treated the same.

Save the program in the file `count.py`

For example:

- for `word="CRAM"` and `phrase="How can a clam cram in a clean cream can?")` the function returns the integer `1`
- for `word="shells"` and `phrase="Sally sells sea shells by the sea shore. But if Sally sells sea shells by the sea shore then where are the sea shells Sally sells?")` the function returns the integer `3`

3. First-last name conventions

In formal writing, instead of "FirstName SecondName ... LastName", we write "LastName, F. S. ...". Write a function `formal(name)` which receives a string `name` with the name in the first form and returns a string with the name in the second form. There's no need to worry about conjunctions such as "de" or "e".

Save the program in the file `formal.py`

For example:

- If the name is **"Aníbal António Cavaco Silva"** the function returns the string **"Silva, A. A. C."** (without quotes)

4. Convert to uppercase

Write a Python function `uppercase(astring)` that receives a string and returns the string with all its characters in uppercase, but only if it contains at least 1 uppercase letter in the first 3 characters~~, that must be letters~~; otherwise the function returns the given string.

Save the program in the file `uppercase.py`

For example:

- for `astring="gin tonic"`, the function returns the string **"gin tonic"** (without quotes)
- for `astring="Gin tonic"`, the function returns the string **"GIN TONIC"** (without quotes)
- for `astring="...tonic..."`, the function returns the string **"...tonic..."** (without quotes)
- for `astring="Κ α λ η μέ ρ α "`, the function returns the string "Κ Α Λ Η ΜΈΡ Α " (without quotes)

5. Number of palindromes in a word

Write a Python function `palindrome(astring)` that receives a string and computes the number of palindromic substrings. A palindrome is a string that reads the same forwards and backwards. Do not count the palindromic substrings with just one character and use the function `format()` to have result as required.

Save the program in the file `palindrome.py`

For example:

- for `astring="geek"`, the function returns the string **"For string 'geek': 1 palindrome substrings"** (without quotes)
- for `astring="ababa"`, the function returns the string **"For string 'ababa': 4 palindrome substrings"** (without quotes)

The end.

FPRO, 2018/19