



Walter Cazzola

Home Page
ADAPT Lab.
Curriculum Vitae
Research Topic

Didactics

Publications

Funded Projects

Research Projects

Related Events



Exam of Advance in Programming

22 July 2013

Disclaimer. Note that to have a running solution for an exercise is not enough: you need a well-cooked solution that proves your ability to use what explained during the classes. The worth of the 2 exercises is 16 and 14 points respectively. To pass the exam you have to do both exercises. The submissions with only one exercise will not be evaluated at all.

Exercise 1: Shakespeare for the Poor.

English (as well as any other natural language) could be really complicated; for example many terms (synonyms) can refer to the same concept but some of them can be more complicated than others and their use can render more complicate to understand the text. This should be evident when you try to read some English literature as Shakespeare.

This exercise consists in writing a function `replace_synonyms` that takes a string (the text to simplify) and returns a copy of the same text with the complicated terms substituted with their synonym more diffuse. To be more evident the new term should be capitalized in the result. The table of the synonym can be downloaded from [here](#) (each line contains the term to be used separated by a colon : from a comma-separated list of terms it will substitute).

The following is an example of the expected computation.

```
if __name__ == "__main__":
    s0 = "The deadline is approximately midnight though it could vary."
    s1 = "She is a fascinating lady; she has an astonishing smile, an alluring voice and an entertaining sense of humor."
    s2 = "The topic is appealing nevertheless the speaker was too monotone"
    print(replace_synonyms(s0))
    print(replace_synonyms(s1))
    print(replace_synonyms(s2))
```

```
[18:31]cazzola@surtur:~/pa/es1>python3 main-simply.py
The deadline is ABOUT midnight BUT it could CHANGE.
She is a INTERESTING lady; she has an AWESOME smile, an BEAUTIFUL voice and an FUNNY sense of humor.
```

Exercise 2: Playing with Strings.

The python module `strings` provides the programmer with several functions to manipulate strings; some functions are really straightforward some others are more complicated.

In this exercise you have to implement your own version of some of these functions, of course without using such a module. In particular the only admitted operations on strings are: '+', '[]', 'in', '==', '!=', and ':'; any solution that doesn't use recursion is considered **wrong** (doesn't matter if shorter or more efficient).

The operations to implement are:

- `reverse(s)`: this returns the string `s` reversed (max 2 points);
- `strip(s, chars)`: this removes all the occurrences of the characters in `chars` from the string `s` and returns the resulting string (max 3 points);
- `split(s, seps)`: this splits the string `s` on the characters listed in `seps` (if two elements of `seps` are adjacent in the string no empty token is generated); the tokens are returned in a list in the order they have in the original string (max 4 points); and
- `find(s, ch)`: this finds the character `ch` in the string `s` and returns the position in the string of the first occurrence of `ch` (positions starts from 0) or -1 when the character doesn't occur in the string; **note** that if the function is called two or more times in a row with the same arguments it returns the position of the next occurrence instead (max 5 points).

The following is an example of the expected computation.

```

from rstrings import *

if __name__ == "__main__":
    s0 = "The deadline is approximately midnight though it could vary."
    s1 = "She is a fascinating lady; she has an astonishing smile, an alluring voice and an entertaining sense of humor."
    s2 = "The topic is appealing nevertheless the speaker was too monotone"
    s3 = "The topic is appealing nevertheless the speaker was too monotone"
    print(strip(s0, 'aeiou'))
    print(reverse(s0))
    print(strip(reverse(s0), 'aeiou'))
    print(split(s1, ' ;,.'))
    print(reverse(s2))
    print("tests on find:")
    print(find(s2, 'a'))
    print(find(s2, 'a'))
    print(find(s2, 'a'))
    print(find(s3, 'a'))
    print(find(s3, 't'))
    print(find(s3, 't'))
    print(find(s3, 't'))
    print(find(s3, 't'))
    print(find(s3, 't'))
    print(find(s3, 't'))

```

```

[18:29]cazzola@surtur:~/pa/es2>python3 main-rstrings.py
Th ddln s pprxmtly mdnght thgh t cld vry.
.yrav dluoc ti hguoht thgindim yletamixorppa si enildaed ehT
.yrv dlc t hght thgndm yltmxrpp s nlld hT
['She', 'is', 'a', 'fascinating', 'lady', 'she', 'has', 'an', 'astonishing',
'smile', 'an', 'alluring', 'voice', 'and', 'an', 'entertaining',
'sense', 'of', 'humor']
.suonotonom oot saw rekaeps eht sselehtreven gnilaepa si cipot ehT
tests on find:
13
17
43
10
4
29
37
53
61
-1

```

Last Modified: Wed, 07 Aug 2013 12:03:34

ADAPT Lab. 