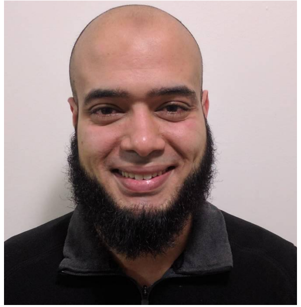
Python Programming

String Homework 2

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Homework 1: Filtering

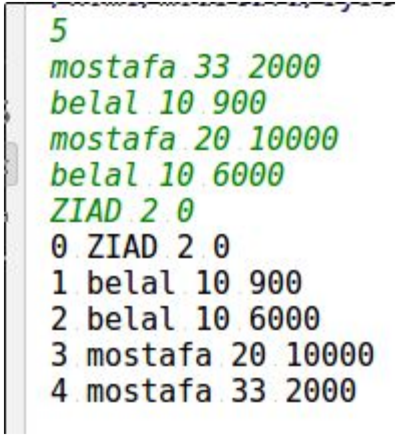
- Read a line of words that could be separated with any of ,$ # (4 letters)
    - Several separators on same line
    - Words themselves have no spaces
- Extract all the words, sort them and print the words
- Input
    - apple,banana, , , apple,student### #student$$apple
- Output:
    - ['apple', 'apple', 'apple', 'banana', 'student', 'student']
- Can you code all of it in a single line? A bit long one
    - Feel free to do it your way

Homework 2: Compressing

- Read a string (sz > 0) of lower letters, then divide it to consecutive groups of the same character and compress them
    - If more than a letter: aaaa ⇒ 4a
    - If a single letter: a ⇒ a
    - Order the groups based on the frequency (**bigger** first), then based on letter (smaller first)
        - Tip: Start with frequency smaller.
- Input ⇒ outputs
    - aaabbbccc ⇒ 3a_3b_3c
    - z ⇒ z
    - aabbbbbddddcccc ⇒ 5b_4c_4d_2a
    - aabbccaa ⇒ 2a_2a_2b_2c

Homework 3: Order Data

- We have N employees in the company, each name, age and salary (both int)
- Line #1: read integer N
- In next N lines: read per line name, age and salary
- Print the data ordered: by name first, then age, then salary
- Input
    - 5
    - mostafa 33 2000
    - belal 10 900
    - mostafa 20 10000
    - belal 10 6000
    - ZIAD 2 0



```
5
mostafa 33 2000
belal 10 900
mostafa 20 10000
belal 10 6000
ZIAD 2 0
0 ZIAD 2 0
1 belal 10 900
2 belal 10 6000
3 mostafa 20 10000
4 mostafa 33 2000
```

Homework 4: Replace substring

- Implement **def** our_replace(main_str, pattern, repalce_with):
    - The function similar to string.replace. We will implement ours
    - Don’t use any method in the string class.
- Read a line of 3 strings
- The function replaces every **pattern** with **to** and return it
    - Input: “aabcabaaad”, “aa”, “x” - Return: “**x**bcab**x**ad”
    - Input: “aabcabaaad”, “aa”, “aaaa” - Return: “**aaaa**bcab**aaaa**ad”
    - Input: “aabcabaaad”, “aa”, “” - Return: “bcabad”
    - Input: “aabcabaaad”, “”, “xx” - Return: “aabcabaaad”

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”