Practice Problem Set 2

Exercise on Strings Level 2

Problem Statement 1

Description:

- Write a Python program that prints a version of the string `s` with all commas replaced by dots.

Expected Output:

String	Output
"Hello, World!"	"Hello. World"
"3,456,344"	"3.456.344"

Problem Statement 2

Description:

- Write a Python program that checks if the string `s` contains all the letters in the alphabet (case-insensitive, so `"A"` should be equivalent to `"a"`).
- If it does, print `True`. Else, print `False`.
- Before comparing the characters, you should convert the string to **lowercase**.
- If the string contains spaces, ignore them before finding the result.
- You may assume that the string doesn't contain any other symbols, only spaces (possibly).
- Consider these letters as part of the alphabet: `'abcdefghijklmnopqrstuvwxyz'`

Expected Output:

String	Output
"abcdefghijklmnopqrstuvwxyz"	True
"The quick brown fox jumps over the lazy dog"	True
"Hello"	False

Hints:

- It is also known as Check **Pangram**
- To use a constant with all letters of the alphabet, you may use `string.ascii_lowercase` from the string module. You can import this module by writing `import string` at the top of your script.
- It can also be helpful to use sets in this problem.

Problem Statement 3

Description:

- Write a Python program that prints a copy of the string `s` without any spaces.
- Words should be connected in the final string.
- If the string doesn't contain spaces, print it intact.

Expected Output:

String	Output
"Hello, World!"	"Hello,World!"
"Have a great day"	"Haveagreatday"
"Python"	"Python"

Problem Statement 4

Description:

- Write a Python program that checks if the string `s` starts with the sequence of characters denoted by the variable `prefix`.
- If it does, print `True`. Else, print `False`.
- This test should be **case sensitive**. For example, `"A"` should **not** be equivalent to `"a"`.
- If the length of the prefix is **greater than** the length of the string, print `False`.

Expected Output:

String	Prefix	Output
"Hello"	"He"	True
"Coding"	"Con"	False
"Nora"	"circum"	False

Problem Statement 5

Description:

- Write a Python program that checks if the string `s` **ends** with a specific sequence of characters denoted by the variable suffix.
- If it does, print `True`. Else, print `False`.
- This test should be case sensitive. Therefore, `"A"` should not be equivalent to `"a"`.
- If the length of the suffix is **greater than** the length of the string, print False.

Expected Output:

String	Suffix	Output
"Hello"	"ello"	True
"Coding"	"eng"	False
"Nora"	"rowing"	False

Problem Statement 6

Description:

- Write a Python program that **reverses** the individual **words** in the string s and **changes** their capitalization. Uppercase letters should be printed in lowercase and vice versa.
- Assume that the string only contains letters and spaces are used to separate words.

Expected Output:

String	Output
"Hello World"	"OLLEh DLROw"
"Python is Awesome"	"NOHTYp SI EMOSEWa"

Hints:

Problem Statement 7

Description:

- Write a Python program to count the number of repeated characters in the string `s`.
- The program must print the total number of repeated characters and a message on the next line displaying the repeated characters separated by a **space** and **sorted** alphabetically.
- If there are no repeated characters in the string, print 0 as the total count and None on the next line.

Expected Output:

String	Output
"Hello"	1 "l"
"Corporation"	2 "or"
"Python"	0 "None"

Hints:

- You might want to keep track of a counter.
- You could store repeated characters in a list. But be careful not to add repeated characters more than once.
- The `sorted()` function returns a sorted version of a list.
- With `print(x, end=" ")`, you can print a sequence of values on the same line.

Problem Statement 8

Description:

- Write a Python program to convert a string `s` to lowercase, sort the characters of each word in alphabetical order, and print the resulting string.
- You may assume that the string only contains letters and spaces to separate the words.
- Spaces should be preserved in the final string.

Expected Output:

String	Output
"Hello World"	"ehllo dlorw"
"Wonderful World"	"deflnoruw dlorw"

Hints:

- In Python, uppercase letters come before lowercase letters in alphabetical order.
- The `sorted()` function can be used to get a sorted list with the characters in a string.