

Aim: Understanding string operations and behavior.

Create a Python project in which you have the following file: **lab8.py**

1. Define a variable `string` and assign the following value to it:

“Madness? This is SPARTA!!!”

- a) Define another variable `string_2` and assign its value in the following way:
Replace the last 3 characters with a dot (‘.’) and assign the result to `string_2`.
Finally, display `string_2`.

- b) Print every word of `string` on a separate line.

2. Define a function `extension` which takes a filename `f_name` as a parameter, including its extension (e.g. “worddocument.docx”) and returns only the extension. If there are no extensions in the given filename, print a corresponding message.

Hint: You can attempt to find the index of the dot character (‘.’) within the filename, and then access all characters after that specific index.

Test your function by calling `extension` twice, once by passing “my_script.py” and once by passing “my_document.docx”.

3. Define a function `occurrence` that takes 2 parameters `source_string` and `letter`. The function should count and display the number of times `letter` appears in `source_string`. It shouldn’t return anything. The argument value to be passed to the parameter `letter` must be a single character, if it’s not, an error message should be given.

Test your function by first getting an input string from the user, and then a letter to be searched within the string. Then, call the function `occurrence` by passing the input string and input character values as arguments.

Note: The argument `letter` can be any character (including punctuation marks, numbers, whitespaces, etc.)

SAMPLE OUTPUT 1 (bold parts are entered by user):

```
Please enter a string to be searched: Mrhaba hocma
Please provide a letter to be searched for: a
The letter 'a' occurs in 'Mrhaba hocma' 3 times.
```

SAMPLE OUTPUT 2 (bold parts are entered by user):

```
Please enter a string to be searched: Abra kadabra alakazam
Please provide a letter to be searched for: abra
Invalid letter!
```

TODO@HOME

- a. Define a function named `print_words` which takes a string as a parameter. The function should automatically detect and print every word found in the string on different lines. The words may be meaningless, but they must be separated from other words with a whitespace character ("space" on your keyboard).

Test your function in the following way: Read a string input from the user and call this function by passing the user input as the argument value.

SAMPLE OUTPUT (bold parts are entered by user):

```
Enter a sentence: This is a sentence.
This
is
a
sentence.
```

- b. Improve `print_words` such that every word is stripped of punctuation marks. Before printing each word, check every character within a word to see if there are any punctuation marks (Check for exclamation and question marks, dots, commas, colons and semicolons). If so, remove them. Assume that the punctuation marks can only be at the end of each word.

SAMPLE OUTPUT (bold parts are entered by user):

```
Enter a sentence: Madness? This is SPARTA!!!
Madness
This
is
SPARTA
```