# Exercise: Text Processing and Regular Expressions

Problems for in-class lab for the: [“Technology Fundamentals” course @ SoftUni](https://softuni.bg/courses/technology-fundamentals).

Submit your solutions in the SoftUni judge system at**:** [Text-Processing-and-Regular-Expressions-Exercise](https://judge.softuni.bg/Contests/1336/Text-Processing-and-Regular-Expressions-Exercise)

## Reveal Words

Write a function, which receives **two parameters**. The first parameter will be a string with some words **splitted by ', '**, parse the given string and collect the words.   
The second parameter will be a string which contains **templates containing '\*'**.  
Find the word with **exact same length** as the **template** and **replace** it.

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 'great',  'softuni is \*\*\*\*\* place for learning new programming languages' | softuni is great place for learning new programming languages |
| 'great, learning',  'softuni is \*\*\*\*\* place for \*\*\*\*\*\*\*\* new programming languages' | softuni is great place for learning new programming languages |

## Modern Times of #(HashTag)

**Find all** special words **starting with #**. Word is invalid if it has **anything other than letters**. Print the words you found without the tag each on a new line.

The input will be **a single string.**

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 'Nowadays everyone uses # to tag a #special word in #socialMedia' | special  socialMedia |

## Palindromes

Write a function, that receives **array of strings**. The function must check for each of the strings, if it's a palindrome or not. If you find a palindrome, print it joined by **', '**.

**If there isn't any palindromes, print "No palindromes found".**

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| [' stella won no wallets',  'not a palindrome'] | stellawonnowallets |
| ['stella won wallets',  'not a palindrome'] | No palindromes found |

## String Substring

Write a program that checks given text for containing a given word. The comparison should be **case insensitive.** Once you find match, **print** the word and **stop** the program. If you don't find the word print **"{word} not found!"**

The input will be given **as two separate strings.**

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 'javascript', 'JavaScript is the best programming language' | javascript |
| 'python',  'JavaScript is the best programming language' | python not found! |

## Emoticon Finder

Find all emoticons in the text. An emoticon **always starts with ":"** and is followed by a **single symbol or letter.**  
The input will be provided as a **single string.**

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 'There are so many emoticons nowadays :P I have many ideas :O what input to place here :)' | :P  :O  :) |

## Pascal-Case Splitter

You will receive a **single** string. This string is written in a **PascalCase format.** Your task here is to split this string by **every word in it**. When you do that log them joined by **comma and space ', '.**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 'SplitMeIfYouCanHaHaYouCantOrYouCan' | Split, Me, If, You, Can, Ha, Ha, You, Cant, Or, You, Can |
| 'HoldTheDoor' | Hold, The, Door |
| 'ThisIsSoAnnoyingToDo' | This, Is, So, Annoying, To, Do |

## Cut and Reverse

In this problem we need to cut the given string **into half** and **reverse** **the two halves.** Print each part on a **separate line.**The input will be **a single string.**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 'tluciffiDsIsihTgnizamAoSsIsihT' | ThisIsDifficult  ThisIsSoAmazing |
| 'sihToDtnaCuoYteBIboJsihTtAdooGoSmI' | IBetYouCantDoThis  ImSoGoodAtThisJob |

## \*Hard Words

You will receive an **array** which holds **string and another array**. The string is a letter from young boy who does not yet know some words and you have to help him. The letter has few **holes**, these holes are the words unknown to the boy and you must fill them with **strings from the array** you receive at the second index. If a hole's **length is 4 you have to replace it with string with the same length** and so on…

### Examples

|  |
| --- |
| **Input** |
| ['Hi, grandma! I\'m so \_\_\_\_ to write to you. \_\_\_\_\_\_ the winter vacation, so \_\_\_\_\_\_\_ things happened. My dad bought me a sled. Mom started a new job as a \_\_\_\_\_\_\_\_\_\_. My brother\'s ankle is \_\_\_\_\_\_\_\_, and now it bothers me even more. Every night Mom cooks \_\_\_ on your recipe because it is the most delicious. I hope this year Santa will \_\_\_\_\_ me a robot.',  ['pie', 'bring', 'glad', 'During', 'amazing', 'pharmacist', 'sprained']] |
| **Output** |
| Hi, grandma! I'm so glad to write to you. During the winter vacation, so amazing things happened. My dad bought me a sled. Mom started a new job as a pharmacist. My brother's ankle is sprained, and now it bothers me even more. Every night Mom cooks pie on your recipe because it is the most delicious. I hope this year Santa will bring me a robot. |

## \*Password Generator

For this problem you have to write a function which generates a password depending on input information. As such, you will be given an **array** of **3 strings.** The first two strings will be at least **10 characters long**, the third one will be **one word.**

Your task here is to concatenate the first two strings, and replace all **vowels** in the **concatenated string** with symbols from the third string. **First vowel** must be replaced with the **first character** from third string, **second vowel** with the **second character** from that string and so on. If the third string is less than the vowels count in the newly formed string you need to start over with **character on 0 index.** When you replace all vowels **reverse** the new password and print it on the console in a format:

**'Your generated password is {password}'**

**Note:** All replaced vowels with the characters from the third string must be upper-case, the rest of the characters are lower-case.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| ['ilovepizza', 'ihatevegetables', 'orange'] | Your generated password is sElbGtNgAvRtOhEGzzNpAvRlO |
| ['easymoneyeazylife', 'atleasttencharacters', 'absolute'] | Your generated password is srTtcUrLhcnOttsSBltAEfTlyzULyOnSmysBA |
| ['areyousureaboutthisone', 'notquitebutitrustyou', 'disturbed'] | Your generated password is SIytsDrtDtEbBtRUqtTnSnIsDhttDEbBRrUsTSyIrD |