# Lab: Text Processing

## Print Characters

Write a function that **receives a** **string** and **prints all the** **characters** on separate lines.

### Input / Output

|  |  |
| --- | --- |
| **Input** | **Output** |
| 'AWord' | A  W  o  r  d |

### Hints

Loop through the string and print each character.



## Remove Occurrences - Иво

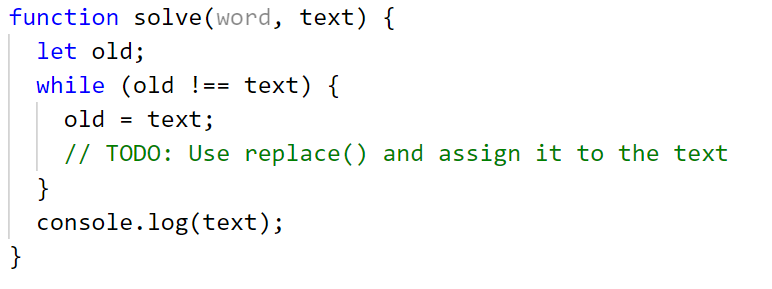
Write a function that receives a **text** and a **word** to remove **all occurrences** of it inside the text.

### Input / Output

|  |  |
| --- | --- |
| **Input** | **Output** |
| ice  kicegiciceeb | kgb |

### Hints

Replace the occurrence of the word inside a **while loop** and use **replace().**



## Substring

Write a function that **receives a string** and **two numbers**. The numbers will be a **starting index** and **count** of elements to substring. Print the result.

### Input / Output

|  |  |
| --- | --- |
| **Input** | **Output** |
| "ASentance", 1, 8 | Sentance |

### Hints

Create a new string that takes the needed amount of elements from the given string.



## Censored Words Иво

Write a function that **receives a text as** a first parameter and a **single word** as a second. Find **all occurrences** of that word in the text and replace them with the corresponding count of '\*'.

### Input / Output

|  |  |
| --- | --- |
| **Input** | **Output** |
| "A small sentence with some words", "small" | A \*\*\*\*\* sentence with some words |

### Hints

Save the new text in a new variable.



The repeat() function should take the length of the word and return that amount of stars '\*'.

## Count String Occurrences

Write a function that **receives a text** and a **string that you need to search**. Print all the occurrences of that word in the string.

### Input / Output

|  |  |
| --- | --- |
| **Input** | **Output** |
| "This is a word and it also is a sentence",  "is" | 2 |

### Hints

Split the sentence into words and create a **counter** that stores how many times the searched word occurs.



## \*\*\*String Iterator

Write a function, wich receives a string and print on the console all the words in it, separated by new line.

Note: **Try to use** [**Iterator Pattern**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Iterators_and_Generators)**!**

|  |  |
| --- | --- |
| **Input** | **Output** |
| "Et cillum do voluptate cillum ut cupidatat aliqua. | Et  cillum  do  voluptate  cillum  ut  cupidatat  aliqua. |

## \*\*\* Password Reset

[Judge Link - https://judge.softuni.bg/Contests/Practice/Index/2303#0](https://judge.softuni.bg/Contests/Practice/Index/2303#0)

*Yet again you have forgotten your password... Naturally it`s not the first time this has happened. Actually you got so tired of it that you decided to help yourself with a smart solution.*

Write a password reset program that performs a series of commands upon a predefined string. First, you will receive a string and afterwards, until the command "**Done**" is given, you will be receiving strings with commands split by a single space. The commands will be the following:

* TakeOdd
  + Takes only the characters at **odd** **indices** and **concatenates** them together to  
    obtain the **new raw password** and then **prints** it.
* Cut {index} {length}
  + Gets the substring with the **given length** starting from the **given index** from the password and removes its first occurrence of it, then prints the password on the console.
  + The given index and length will **always** be **valid**.
* Substitute {substring} {substitute}
  + If the raw password contains the given substring, replaces all of its   
    occurrences with the substitute text given and prints the result.
  + If it doesn’t, prints "Nothing to replace!"

### Input

* You will be receiving strings until the "**Done**" command is given.

### Output

* After the "Done" command is received, print:
  + "Your password is: {password}"

### Constraints

* The indexes from the "**Cut {index} {length}**" command will always be valid.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Siiceercaroetavm!:?:ahsott.:i:nstupmomceqr  TakeOdd  Cut 15 3  Substitute :: -  Substitute | ^  Done | icecream::hot::summer  icecream::hot::mer  icecream-hot-mer  Nothing to replace!  Your password is: icecream-hot-mer |
| **Comments** | |
| TakeOdd  Siiceercaroetavm!:?:ahsott.:i:nstupmomceqr -> icecream::hot::summer  We only take the chars at odd indices 1, 3, 5 etc.  **Cut 15 3 ->** icecream::hot::summer -> sum  icecream::hot::mer  We cut a substring starting at index 15 with length 3,  remove it from the raw password and print it.  Then, on a new line we print the resulting new raw password.  **Substitute :: - ->** icecream::hot::summer -> icream-hot-summer  We replace "::" with "-".  **Substitute** | ^ **->** Nothing to replace!  "|" is not found anywhere in the raw password.  Finally, after receiving the "**Done**" command, we print the resulting password in the proper format. | |
| **Input** | **Output** |
| up8rgoyg3r1atmlmpiunagt!-irs7!1fgulnnnqy  TakeOdd  Cut 18 2  Substitute ! \*\*\*  Substitute ? .!.  Done | programming!is!funny  programming!is!fun  programming\*\*\*is\*\*\*fun  Nothing to replace!  Your password is: programming\*\*\*is\*\*\*fun |