# Lab: Strings and RegEx

This document defines the homework assignments from the ["Programming Fundamentals" Course @ Software University](https://softuni.bg/courses/programming-fundamentals). Please submit your solutions (source code) of all below described problems in [Judge](https://judge.softuni.bg/Contests/320/Strings-and-RegEx-Lab).

# String operations

## Reverse string

Write a program that reads a string from the console, reverses it and prints the result back at the console.

|  |  |
| --- | --- |
| **Input** | **Output** |
| sample | elpmas |
| 24tvcoi92 | 29iocvt42 |

## Count substring occurrences

Write a program to **find how many times a given string appears in a given text as substring**. The text is given at the first input line. The search string is given at the second input line. The output is an integer number. Please ignore the **character casing**. **Overlapping** between occurrences is **allowed**. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| **Wel**come to the Software University (SoftUni)! **Wel**come to programming. Programming is **wel**lness for developers, said Max**wel**l.  wel | 4 |
| **aaaaaa**  aa | 5 |
| **ababa** c**aba**  aba | 3 |
| Welcome to SoftUni  Java | 0 |

## Text filter

Write a program that takes a **text** and a **string of banned words**. All words included in the ban list should be replaced with **asterisks** "**\***", equal to the word's length. The entries in the ban list will be separated by a **comma** and **space** "**,** ".

The ban list should be entered on the first input line and the text on the second input line. Example:

|  |  |
| --- | --- |
| **Input** | **Output** |
| Linux, Windows  It is not **Linux**, it is GNU/**Linux**. **Linux** is merely the kernel, while GNU adds the functionality. Therefore we owe it to them by calling the OS GNU/**Linux**! Sincerely, a **Windows** client | It is not \*\*\*\*\*, it is GNU/\*\*\*\*\*. \*\*\*\*\* is merely the kernel, while GNU adds the functionality. Therefore we owe it to them by calling the OS GNU/\*\*\*\*\*! Sincerely, a \*\*\*\*\*\*\* client |

## Palindromes

Write a program that extracts from a given text all palindromes, e.g. ABBA, lamal, exe and prints them on the console on a single line, separated by comma and space. Use spaces, commas, dots, question marks and exclamation marks as word delimiters. Print only **unique** palindromes, **sorted** lexicographically.

Example:

|  |  |
| --- | --- |
| **Input** | **Output** |
| Hi,exe? ABBA! Hog fully a string. Bob | a, ABBA, exe |

# Playing with RegEx

## Match full name

Write a regular expression to match a valid full name. A valid full name consists of **two words**, each word **starts** with a **capital letter** and contains **only lowercase letters afterwards**; each word should be **at least two letters long**; the two words should be **separated by a single space**.

To help you out, we've outlined several steps:

* Use an online regex tester like <https://regex101.com/>
* Check out how to use **character sets** (denoted with square brackets - "[]")
* Specify that you want two words with a space between them (the **space character ' '**, and not any whitespace symbol)
* For each word, specify that it should begin with an uppercase letter using a **character set**. The desired characters are in a range – **from 'A' to 'Z'**.
* For each word, specify that what follows the first letter are only **lowercase letters**, one or more – use another character set and the correct **quantifier**.
* To prevent capturing of letters across new lines, put "\b" at the beginning and at the end of your regex. This will ensure that what precedes and what follows the match is a word boundary (like a new line).

In order to check your regex, use these values for reference (paste all of them in the Test String field):

|  |  |
| --- | --- |
| **Match ALL of these** | **Match NONE of these** |
| Ivan Ivanov | ivan ivanov, Ivan ivanov, ivan Ivanov, IVan Ivanov, Ivan IvAnov, Ivan Ivanov |

## Match phone number

Write a regular expression to match a valid phone number from Sofia. A valid number will start with "+359" followed by the area code (2) and then the number itself, consisting of 7 digits (separated in two group of 3 and 4 digits respectively). The different parts of the number are separated by **either a space or a hyphen** ('-'). Refer to the examples to get the idea.

* Use quantifiers to match a specific number of digits
* Use a capturing group to make sure the delimiter is **only one of the allowed characters** **(space or hyphen)** and not a combination of both. Use the group number to achieve this
* Add a word boundary at the end of the match to avoid partial matches (the last example on the right-hand side)
* Ensure that before the '+' sign there is either a space or the beginning of the string

|  |  |
| --- | --- |
| **Match ALL of these** | **Match NONE of these** |
| +359 2 222 2222  +359-2-222-2222 | 359-2-222-2222, +359/2/222/2222, +359-2 222 2222  +359 2-222-2222, +359-2-222-222, +359-2-222-22222 |

# Build-in RegEx

## Replace <a> tag

Write a programthat replaces in a HTML document given as string **all the tags <a href=…>…</a>** with corresponding **tags [URL href=…>…[/URL]**.Read an input, until you receive **“end” command**. Print the result on the console.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| <ul> <li> <a href=**"http://softuni.bg"**>SoftUni</a>  </li> </ul>  **end** | <ul> <li>  [URL href=**"http://softuni.bg"**>SoftUni[/URL]  </li>  </ul>  </li> </ul> |

**Note**: The input may be read on a single line (unlike the example above) or from a file. Remove all new lines if you choose the first approach.