# Homework: Text Processing and Regex API

This document defines homework assignments from the [“Java Basics“ Course @ Software University](https://softuni.bg/courses/java-basics/). Please submit as homework a single zip / rar / 7z archive holding the solutions (source code) of all below described problems.

## Extract Emails

Write a program to **extract all email addresses from given text**. The text comes at the first input line. Print the emails in the output, each at a separate line. Emails are considered to be in format **<user>@<host>**, where:

* **<user>** is a sequence of letters and digits, where '**.**', '**-**' and '**\_**' can appear between them. Examples of valid users: "**stephan**", "**mike03**", "**s.johnson**", "**st\_steward**", "**softuni-bulgaria**", "**12345**". Examples of invalid users: ''**--123**", ".....", "**nakov\_-**", "**\_steve**", "**.info**".
* **<host>** is a sequence of at least two words, separated by dots '**.**'. Each word is sequence of letters and can have hyphens '**-**' between the letters. Examples of hosts: "**softuni.bg**", "**software-university.com**", "**intoprogramming.info**", "**mail.softuni.org**". Examples of invalid hosts: "**helloworld**", "**.unknown.soft.**", "**invalid-host-**", "**invalid-**".
* Example of **valid emails**: info@softuni-bulgaria.org, kiki@hotmail.co.uk, no-reply@github.com, s.peterson@mail.uu.net, info-bg@software-university.software.academy.

Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| Please contact us at: support@github.com. | support@github.com |
| Just send email to s.miller@mit.edu and j.hopking@york.ac.uk for more information. | s.miller@mit.edu  j.hopking@york.ac.uk |
| Many users @ SoftUni confuse email addresses. We @ Softuni.BG provide high-quality training @ home or @ class. –- steve.parker@softuni.de. | steve.parker@softuni.de |

## Count Substring Occurrences

Write a program to **find how many times given string appears in 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. 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 |

## Count Specified Word

Write a program to **find how many times a word appears in given text**. The text is given at the first input line. The target word is given at the second input line. The output is an integer number. Please ignore the character casing. Consider that any non-letter character is a word separator. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| Welcome to the Software University (SoftUni)! Welcome to programming.  welcome | 2 |
| I am coming...  hello | 0 |
| It's OK, I'm in.  i | 1 |
| Java is a set of several computer software products and specifications from Oracle Corporation that provides a system for developing application software and deploying it in a cross-platform computing environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones on the low end, to enterprise servers and supercomputers on the high end.  is | 2 |

## Count All Words

Write a program to **count the number of words** in given sentence. Use any non-letter character as word separator.

Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| Welcome to the Software University (SoftUni)! | 6 |
| I am coming... | 3 |
| It's OK, I'm in. | 6 |
| Java is a set of several computer software products and specifications from Oracle Corporation that provides a system for developing application software and deploying it in a cross-platform computing environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones on the low end, to enterprise servers and supercomputers on the high end. | 60 |

## Extract words

Write a program that extracts words from a string. Words are sequences of characters that are at least two symbols long and consist **only** of English alphabet letters. **Use regex.**

|  |  |
| --- | --- |
| **Words** | **Output** |
| Az&76sym&&samo&cvqt&lilav | Az sym samo cvqt lilav |
| Shoot18297the1231023dwarves! | Shoot the dwarves |
| 1798No(\*&Girls)\*(09Allowed | No Girls Allowed |

## Starts and Ends With Capital Letter

Write a program that takes as input an array of strings are prints only the words that start and end with capital letter. Words are only strings that consist of English alphabet letters. **Use regex.**

|  |  |
| --- | --- |
| **Words** | **Output** |
| GoshO blabla NqmaSm1saL KvoStaA | GoshO KvoStaA |
| AZ AK 47 RoBoT noWayouT | AZ AK RoBoT |
| DrakonI Navsekyde | DrakonI |

## \*\*\*Magic exchangeable words

Write a **method** that takes as input two strings of equal length, and returns Boolean if they are exchangeable or not. Exchangeable are words where the characters in the first string can be replaced to get the second string. Example: **egg** and **add** are exchangeable, but **aabbccbb** and **nnooppzz** are not. (First **bb** corresponds to **oo**, but second **bb** corresponds to **zz**)

|  |  |
| --- | --- |
| **Input** | **Output** |
| gosho hapka | true |
| aabbaa ddeedd | true |
| foo bar | false |