# Lab: Text Processing

Please, submit your source code solutions for the described problems to the [Judge System](https://judge.softuni.org/Contests/1739/Text-Processing-Lab).

## Reverse Strings

You will be given strings on separate lines until you receive an **"end"** command. Write a program that reverses strings and prints each pair on a separate line in the format **"{word} = {reversed\_word}"**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| helLo  Softuni  bottle  end | helLo = oLleh  Softuni = inutfoS  bottle = elttob |
| Dog  caT  chAir  end | Dog = goD  caT = Tac  chAir = riAhc |

### Hint

Read the first line and create a while loop until you receive "**end**":



Now loop back through the text and add each character to a variable that stores the reversed string:



## Repeat Strings

Write a program that reads a **sequence of strings**,separated by a single space. Each string should be repeated N times, where **N** is the **length of the string**. Print the final strings concatenated into one string.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| hi abc add | hihiabcabcabcaddaddadd |
| work | workworkworkwork |
| ball | ballballballball |

### Hint

Start by reading the input and splitting it (we can receive multiple words):



Loop through each word, get its length, and add the repeated word to the result:



## Substring

On the **first line,** you will receive a **string**. On the **second line,** you will receive a second **string**. Write a program that **removes** **all** the **occurrences** of the **first** string **in** the **second** **until** there is **no match**. At the end, **print** the **remaining string**.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comment** |
| ice  kicegiciceeb | kgb | We remove ice once and we get "kgiciceeb"  We match "ice" one more time and we get "kgiceb"  There is one more match. The finam result is "kgb" |

### Hints

First, read the two lines:



Create a while loop that replaces all occurrences of the string with an empty string and print the result:



## Text Filter

Write a program that receives a **text** and a **string of banned words**, separated by a **comma** and **space** **", "**. All banned words in the text should be replaced with the number of **asterisks** **"\*"**, equal to the word's length.

The ban list will be entered on the **first** input line and the text - on the **second** input line.

### Examples

|  |  |
| --- | --- |
| **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 |

### Hints

Read the banned words and the text:



Loop through all the words and replace them with "**\***" while they are in the text:



## Digits, Letters, and Other

Write a program that receives a **single** **string**. On the **first** **line,** print **all the digits found in the string**, on the **second** – **all the letters**, and on the **third** – **all the other characters**. There will **always** be at least one digit, one letter, and one other character.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Agd#53Dfg^&4F53 | 53453  AgdDfgF  #^& |

### Hints

Read the input and create 3 empty strings for each type:



Loop through each character and check if it is a digit, a letter, or other:



Print the result:

