# Lab: Text Processing

Problems for in-class lab for the [Python Fundamentals Course @SoftUni](https://softuni.bg/trainings/3368/python-fundamentals-may-2021).

Submit your solutions in the SoftUni judge system at <https://judge.softuni.bg/Contests/1739>.

## Reverse Strings

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

word = input()  
  
while word != "end":  
 mirror\_word = ''  
 for x in reversed(word):  
 mirror\_word += x  
 print(f"{word} = {mirror\_word}")  
  
 word = input()

### Examples

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

### Hint

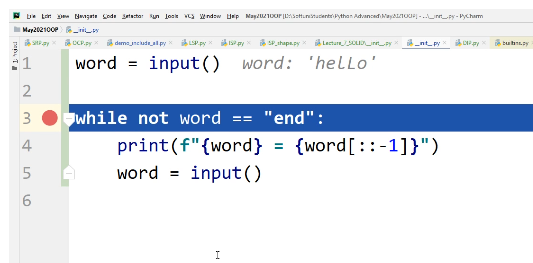
Read the first line and create a while loop until you receive "**end**":



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



Another solution with slicing:



## Repeat Strings

Write a program which 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.

input\_string = input().split()  
  
for x in input\_string:  
 sub\_string = x  
 print(sub\_string \* len(sub\_string), end='')

### 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 which **removes** **all** the **occurrences** of the **first** string **in** the **second** **until** there is **no match**. At the end **print** the **remaining string**.

string\_fragment = input()  
text = input()  
  
while string\_fragment in text:  
 text = text.replace(string\_fragment, '')  
  
print(text)

### 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 which replaces all occurrences of the string with empty string and print the result:



## Text Filter

Write a program which receives a **text** and a **string of banned words**, separated by a **comma** and **space** **", "**. All words included in the ban list should be replaced with **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.

banned\_words = input().split(', ')  
text = input()  
  
for x in banned\_words:  
 while x in text:  
 text = text.replace(x, "\*" \* len(x))  
  
print(text)

### 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 which receives a **single** **string**. On the **first** **line** it should 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 characters.

### 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 digit, letter or other:



Print the result:



string\_all\_chars = input()  
  
digits = []  
letters = []  
others = []  
  
for x in string\_all\_chars:  
 if x.isdigit():  
 digits.append(x)  
 elif x.isalpha():  
 letters.append(x)  
 else:  
 others.append(x)  
  
print(''.join(digits))  
print(''.join(letters))  
print(''.join(others))