

More Exercises: Text Processing

Additional exercises for the [Python Fundamentals Course @SoftUni](#).
Submit your solutions in the SoftUni judge system at <https://judge.softuni.org/Contests/1741>.

Note: All the exercises are excluded from your homework!

1. Extract Person Information

Write a program that reads **N** lines of strings and extracts the **name** and the **age** of a given person:

- The person's name will be **surrounded by "@"** and **"|"** in the format **"@{name}|"**.
- The person's **age** will be **surrounded by "#"** and **"*"** in the format **"#{age}*"**.

Example: "Hello my name is @Peter| and I am #20* years old."

For each found name-age pair, **print** a line in the following format **"{name} is {age} years old."**

Example

Input	Output
2 Here is a name @George and an age #18* Another name @Billy #35* is his age	George is 18 years old. Billy is 35 years old.
3 random name @lilly random digits #5*age @Marry with age #19* here Comes @Garry he is #48* years old	lilly is 5 years old. Marry is 19 years old. Garry is 48 years old.

2. ASCII Sumator

Write a program that prints the **sum of all found characters between two given characters** (their **ASCII code**). On each of the **first two lines**, you will receive a single **character**. On the **last line**, you get a **random string**. **Print the sum of the ASCII values of all characters** in the random string between the two given characters in the ASCII table.

Example

Input	Output	
. @ dsg12gr5653feee5	363	The found characters are 1, 2, 5, 6, 5, 3 and 5. Their ASCII sum is 49 + 50 + 53 + 54 + 53 + 51 + 53 = 363.
? E @ABCEF	262	

3. Treasure Finder

Write a program that **decrypts a message** by a given **key** and gathers information about hidden **treasure type** and its **coordinates**. On the **first line**, you will receive a **key (sequence of numbers separated by a space)**. On the **next few lines**, you will receive lines with strings until you get the command **"find"**.

You should **loop through every string** and **decrease the ASCII code of each character** with a **corresponding number of the key** sequence. You choose a key number from the sequence by just **looping through it**. If the **length of the key** sequence is **less than the string** sequence, you should continue **looping from the beginning**.

For more clarification, see the example below.

After decrypting the message, you will **get a type of treasure and its coordinates**. The **type** will be **between** the symbol "&", and the coordinates - between the symbols "<" and ">".

For each line **print the type and the coordinates** in the format **"Found {type} at {coordinates}"**.

Example

Input	Output	Comment
1 2 1 3 ikegfp'jpne)bv=41P83X@ ujfufKt)Tkmyft'duEprsfjqbfv=53V55XA find	Found gold at 10N70W Found Silver at 32S43W	We start looping through the first string and the key. When we reach the end of the key, we start looping from the beginning of the key, but we continue looping through the string. (until the string is over) The first message is: "hidden&gold&at<10N70W>" so we print we found gold at the given coordinates We do the same for the second string "therels&Silver&atCoordinates<32S43W>" (starting from the beginning of the key and the beginning of the string)

4. Morse Code Translator

Write a program that translates messages from **Morse code to English (capital letters)**. Use [this](#) page to help you (**without the numbers**). The words will be separated by a **space (' ')**. Each word is separated by a **' | '**.

Print the **found words** on one line, separated by a **space**.

Example

Input	Output
.. -- .- -.. . -.- -.- ..- .- .- . - . .- .-. -.- .- -.- -.- -.- .. .	I MADE YOU WRITE A LONG CODE
.. -.- .- . -.- -.- ..- .- .- . -. -.- - -.- .- .-	I HOPE YOU ARE NOT MAD

5. HTML

You will receive **lines** of input:

- On the **first line**, you will receive a **title of an article**
- On the **second line**, you will receive the **content of that article**
- On the following lines, until you receive **"end of comments"** you will get the **comments about the article**

Print the **whole information in html format**:

- The **title** should be in "**h1**" tag (**<h1></h1>**)
- The **content** in **article tag** (**<article></article>**)
- Each **comment** should be in **div tag** (**<div></div>**)

For more clarification see the example below.

Example

Input	Output
SoftUni Article Some content of the SoftUni article some comment more comment last comment end of comments	<pre><h1> SoftUni Article </h1> <article> Some content of the SoftUni article </article> <div> some comment </div> <div> more comment </div> <div> last comment </div></pre>