Programming exercise:

# Login

Write a program that recognizes the following users:

| username | password |
| --- | --- |
| alex | sunshine |
| emma | haskell |

The program either shows a personal message or informs of an incorrect username or password.

Sample output

```
Enter username: alex
Enter password: sunshine
You have successfully logged in!
```

Sample output

```
Enter username: emma
Enter password: haskell
You have successfully logged in!
```

Sample output

```
Enter username: alex
Enter password: thunderstorm
Incorrect username or password!
```

**NB!** You can't compare strings with `==`!

**NB!** Logins should not be implemented like this in real life! You can become familiar with safer ways to implement logins on courses focusing on web programming.

Programming exercise:

# Line by line

Write a program that reads strings from the user. If the input is empty, the program stops reading input and halts. For each non-empty input it splits the string input by whitespaces and prints each part of the string on a new line.

Sample output

```
once upon a time
once
upon
a
time
a little program
a
little
program
halted
halted
```

Programming exercise:

# AV Club

Write a program that reads user input until an empty line. For each non-empty string, the program splits the string by spaces and then prints the pieces that contain `av`, each on a new line.

Sample output

```
java is a programming language
java
navy blue shirt
navy
```

Sample output

```
Do you have a favorite flavor
have
favorite
flavor
was it a cat?
```

Tip! Strings have a `contains`-method, which tells if a string contains another string. It works like this:

```
String text = "volcanologist";

if (text.contains("can")) {
    System.out.println("can was found");
}

if (!text.contains("tin")) {
    System.out.println("tin wasn't found");
}
```

Sample output

```
can was found
tin wasn't found
```

Programming exercise:

# First words

Write a program that reads user input until an empty line is entered. For each non-empty line the program splits the string by spaces | prints the first part of the string.

Sample output

```
one two three four
one
this is a very important message
this
```

Programming exercise:

# LastWords

Write a program that reads user input until an empty line is entered. For each non-empty line the program splits the string by spaces | prints the last part of the string.

Sample output

```
one two three four
four
this is a very important message
message
```

Tip! You can find out the length of the array like this:

```
String[] parts = {"one", "two", "three"};
System.out.println("Number of parts: " + parts.length);
```

Sample output

```
Number of parts: 3
```

Programming exercise:

# Age of the oldest

Write a program that reads names and ages from the user until an empty line is entered. The name and age are separated by a comma.

After reading all user input, the program prints the age of the oldest person. You can assume that the user enters at least one person, and that one of the users is older than the others.

Sample output

```
sebastian,2
lucas,2
lily,1
hanna,5
gabriel,10
Age of the oldest: 10
```

Programming exercise:

# Name of the oldest

Write a program that reads names and ages from the user until an empty line is entered. The name and age are separed by a comma.

After reading all user input, the program prints the name of the oldest person. You can assume that the user enters at least one person, and the that one of the users is older than the others.

Sample output

```
sebastian,2
lucas,2
lily,1
hanna,5
gabriel,10

Name of the oldest: gabriel
```

Programming exercise:

# Personal details

Write a program that reads names and birth years from the user until an empty line is entered. The name and birth year are separated by a comma.

After that the program prints the longest name and the average of the birth years. If multiple names are equally longest, you can print any of them. You can assume that the user enters at least one person.

Sample output

```
sebastian,2017
lucas,2017
lily,2017
hanna,2014
gabriel,2009

Longest name: sebastian
Average of the birth years: 2014.8
```

Sample output

```
sauli,1948
tarja,1943
martti,1936
mauno,1923
urho,1900

Longest name: martti
Average of the birth years: 1930.0
```