Lab: Functional Programming

This document defines the lab for the "Java Advanced" course @ Software University. Please submit your solutions (source code) to all below-described problems in Judge.

1. Sort Even Numbers

Write a program that reads one line of **Integers** separated by **", "**.

- Print the **even** numbers.
- **Sort** them in ascending order.
- Print them again.

Use 2 **Lambda Expresions** to do so.

Examples

Input	Output
4, 2, 1, 3, 5, 7, 1, 4, 2, 12	4, 2, 4, 2, 12 2, 2, 4, 4, 12
1, 3, 5	*(no output)*
2, 4, 6	2, 4, 6 2, 4, 6

Hints

- It is up to you what type of data structures you will use to solve this problem.
- Try different ways, of solving this problem, for example:

```
numbers.removeIf(num -> num % 2 != 0);
numbers.sort(Integer::compareTo);
```

2. Sum Numbers

Write a program that reads one line of **Integers** separated by **", "**. Print the **count** of the numbers and their **sum.**

Use a **`Function<String, Integer>.`**

Examples

Input	Output
4, 2, 1, 3, 5, 7, 1, 4, 2, 12	Count = 10 Sum = 41
2, 4, 6	Count = 3 Sum = 12

Hints

- Use **`Function<String, Integer>`** for **parsing** integers after you split them into a **String** array.

3. Count Uppercase Words

Write a program that reads one line of **text** from the console. Print the **count** of words that start with an **Uppercase letter**, after that print all these **words** in the **same order** as you found them in the text.

Use a **`Predicate<String>`.**

Examples

Input	Output

`The following example shows how to use Predicate`	`2`
	`The`
	`Predicate`

`Write a program that reads one line of text from console. Print count of words that start with Uppercase, after that print all those words in the same order like you find them in text.`	`3`
	`Write`
	`Print`
	`Uppercase,`

Hints

- Use a **`Predicate<String>`** like an **if-condition.**

4. Add VAT

Write a program that reads one line of **Double** prices separated by **", "**. Print the prices with added **VAT**s for all of them. Format them to the **2nd** digit after the decimal point. The order of the prices must remain the same.

Use an **`UnaryOperator<Double>`.**

Examples

Input	Output

`1.38, 2.56, 4.4`	`Prices with VAT:`
	`1.66`
	`3.07`
	`5.28`

`1, 3, 5, 7`	`Prices with VAT:`
	`1.20`
	`3.60`
	`6.00`
	`8.40`

Hints

```
UnaryOperator<Double> addVat = value -> value * 1.2;
//TODO: Foreach price:
double priceWithVAT = addVat.apply(price);
```

5. Filter by Age

Write a program that reads an integer **N** on the first line. And on next **N** lines read pairs of **"{name}, {age}".** Then read three more lines:

- Condition - "**younger**" or "**older**"
- Age - **Integer**
- Format - "**name**", "**age**" or "**name age**"

Depending on the **condition,** print the **pairs** in the right **format**.

Don't use any built-in functionality. Write your methods.

Examples

Input	Output
5 Peter, 20 George, 18 Maria, 29 Idan, 31 Simeon, 16 older 20 name age	Peter - 20 Maria - 29 Idan - 31

Input	Output
5 Peter, 20 George, 18 Maria, 29 Idan, 31 Simeon, 16 younger 20 name	Peter George Simeon

Input	Output
5 Peter, 20 George, 18 Maria, 29 Idan, 31 Simeon, 16 younger 50 age	20 18 29 31 16

6. Find Evens or Odds

You are given a **lower** and an **upper bound** for a range of integer numbers. Then a command specifies if you need to list all **even or odd** numbers in the given range. Use **predicates** that need to be **passed to a method.**

Examples

Input	Output
1 10 odd	1 3 5 7 9
20 30 even	20 22 24 26 28 30