# Lab: Functional Programming

## Sort Even Numbers

Write a program that reads one line of **integers** separated by **", "**. Then prints the **even numbers** of that sequence **sorted** in **increasing** order.

### Examples

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 4, 2, 1, 3, 5, 7, 1, 4, 2, 12 | 2, 2, 4, 4, 12 | 1, 3, 5 |  | 2, 4, 6 | 2, 4, 6 |

### Hint

It is up to you what type of data structures you will use to solve this problem. Use functional programming filter and sort the collection of numbers.

## Sum Numbers

Write a program that reads a line of **integers** separated by **", "**. Print on two lines the **count** of numbers and their **sum.**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4, 2, 1, 3, 5, 7, 1, 4, 2, 12 | 10  41 |
| 2, 4, 6 | 3  12 |

## Count Uppercase Words

Write a program that reads a line of **text** from the console. Print **all** the words that start with an **uppercase letter** in the **same order** you've received them in the text.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| The following example shows how to use Function | The  Function |
| 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. | Write  Print  Uppercase, |

### Hint

Use **Func<string, bool>** and use **" "** for splitting words.

## Add VAT

Write a program that reads one line of **double** prices separated by **", "**. Print the **prices** with **added** **VAT** for all of them. **Format** them to **2** **signs** after the decimal point. The **order** of the prices must be the **same**.  
VAT is equal to 20% of the price.

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 1.38, 2.56, 4.4 | 1.66  3.07  5.28 | 1, 3, 5, 7 | 1.20  3.60  6.00  8.40 |

## Filter by Age

Write a program that receives an integer **N** on first line. On the next **N** lines, read pairs of **"[name], [age]".** Then read three lines with:

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

Depending on the **condition**, print the correct **pairs** in the correct **format**. **Don’t use the built-in functionality from .NET. Create your own methods.**

### Examples

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 5  Lucas, 20 Tomas, 18 Mia, 29 Noah, 31 Simo, 16  older  20  name age | Lucas - 20  Mia - 29  Noah - 31 | 5  Lucas, 20 Tomas, 18 Mia, 29 Noah, 31 Simo, 16  younger  20  name | Tomas  Simo |  | 5  Lucas, 20 Tomas, 18 Mia, 29 Noah, 31 Simo, 16  younger  50  age | 20  18  29  31  16 |