# Exercises: Regular Expressions

Problems for more exercise for the ["PHP Fundamentals" course @ SoftUni](https://softuni.bg/trainings/2344/php-fundamentals-may-2019).

You can check your solutions in [Judge](https://judge.softuni.bg/Contests/1755/).

## Star Enigma

The war is in its peak, but you, young Padawan, can turn the tides with your programming skills. You are tasked to create a program to **decrypt** the messages of The Order and prevent the death of hundreds of lives.

You will receive several messages, which are **encrypted** using the legendary star enigma. You should **decrypt the messages**, following these rules:

To properly decrypt a message, you should **count all the letters** **[s, t, a, r]** – **case insensitive** and **remove** the count from the **current ASCII value of each symbol** of the encrypted message.

After decryption:

Each message should have a **planet name, population, attack type ('A', as attack or 'D', as destruction) and soldier count.**

The planet name **starts after** **'@'** and contains **only letters from the Latin alphabet**.

The planet population **starts after ':'** and is an **Integer**;

The attack type may be **"A"(attack) or "D"(destruction)** and must be **surrounded by "!"** (exclamation mark).

The **soldier count** starts after **"->"** and should be an Integer.

The order in the message should be: **planet name -> planet population -> attack type -> soldier count.** Each part can be separated from the others by **any character except: '@', '-', '!', ':' and '>'.**

### Input / Constraints

* The **first line** **holds n** – the number of **messages**– **integer in range [1…100];**
* On the next **n** lines, you will be receiving encrypted messages.

### Output

After decrypting all messages, you should print the decrypted information in the following format:

First print the attacked planets, then the destroyed planets.  
"Attacked planets: {attackedPlanetsCount}"  
"-> {planetName}"  
"Destroyed planets: {destroyedPlanetsCount}"  
"-> {planetName}"

The planets should be **ordered by name** **alphabetically.**

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 2  **ST**CDoghudd4=63333$D$0**A**53333  EHf**s**y**ts**nhf?8555&I&2C9555**SR** | Attacked planets: 1  -> Alderaa  Destroyed planets: 1  -> Cantonica | We receive two messages, to decrypt them we calculate the key:  First message has decryption key 3. So we substract from each characters code 3.  **PQ@Alderaa1:30000!A!->20000**  The second message has key 5.  **@Cantonica:3000!D!->4000NM**  **Both messages are valid** and they contain planet, population, attack type and soldiers count.  After decrypting all messages we print each planet according the format given. |
| **Input** | **Output** | **Comments** |
| 3  **tt**(''DG**s**vywge**r**x>6444444444%H%1B9444  GQh**rr**|**A**977777(H(**TTTT**  EHf**s**y**ts**nhf?8555&I&2C9555**SR** | Attacked planets: 0  Destroyed planets: 2  -> Cantonica  -> Coruscant | We receive three messages.  Message one is decrypted with key 4:  **pp$##@Coruscant:2000000000!D!->5000**  Message two is decrypted with key 7:  **@Jakku:200000!A!MMMM**  This is **invalid message**, missing soldier count, so we continue.  The third message has key 5.  **@Cantonica:3000!D!->4000NM** |

"It’s a trap!" – Admiral Ackbar

## Nether Realms

Mighty battle is coming. In the stormy nether realms, demons are fighting against each other for supremacy in a duel from which only one will survive.

Your job, however is not so exciting. You are assigned to **sign in all the participants** in the nether realm's mighty battle's demon book, which of course is **sorted alphabetically**.

A demon's **name contains his health and his damage**.

The **sum of the asci codes** of **all characters** (excluding numbers (0-9), arithmetic symbols (**'+', '-', '\*', '/'**) and delimiter dot (**'.'**)) gives a **demon's total health**.

**The sum of all numbers** in his name forms his base damage. Note that you should consider the plus **'+'** and minus **'-'** signs (e.g. **+10 is 10** and **-10 is -10**). However, there are some symbols (**'\*'** and **'/'**) that can further **alter the base damage by multiplying or dividing it by 2** (e.g. in the name "m**15**\*/c**-5.0**", the base damage is **15 + (-5.0) = 10** and then you need to multiply it by 2 (e.g. 10 \* 2 = 20) and then divide it by 2 (e.g. 20 / 2 = 10)).

So, **multiplication and division** are applied **only after all numbers are included** in the calculation and **in the order they appear in the name**.

You will get all demons **on a single line**, separated by commas and zero or more blank spaces. Sort them in **alphabetical order** and print their names **along their health and damage**.

### Input

The input will be read from the console. The input consists of a **single line** containing all demon names **separated by commas and zero or more spaces** in the format: **"{demon name}, {demon name}, … {demon name}"**

### Output

Print all demons **sorted by their name in ascending order**, each on a separate line in the format:

* **"{demon name} - {health points} health, {damage points} damage"**

### Constraints

* A demon's name will contain **at least one character**
* A demon's name **cannot contain** blank spaces ' ' or commas ','
* A **floating point number will always have digits before and after its decimal separator**
* **Number** in a demon's name **is considere**d everything that is a valid integer or floating point number (with dot '.' used as separator). For example, all these are valid numbers: '4', '+4', '-4', '3.5', '+3.5', '-3.5'

### Examples

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Output** | **Comments** | |
| M3ph-0.5s-0.5t0.0\*\* | M3ph-0.5s-0.5t0.0\*\* - 524 health, 8.00 damage | M3ph-0.5s-0.5t0.0\*\*:  Health = 'M' + 'p' + 'h' + 's' + 't' = 524 health.  Damage = (3 + (-0.5) + (-0.5) + 0.0) \* 2 \* 2 = 8 damage. | |
| **Input** | **Output** | | **Comments** |
| M3ph1st0\*\*, Azazel | Azazel - 615 health, 0.00 damage  M3ph1st0\*\* - 524 health, 16.00 damage | | Azazel:  Health - 'A' + 'z' + 'a' + 'z' + 'e' + 'l' = 615 health. Damage - no digits = 0 damage.  M3ph1st0\*\*:  Health - 'M' + 'p' + 'h' + 's' + 't' = 524 health.  Damage - (3 + 1 + 0) \* 2 \* 2 = 16 damage. |
| Gos/ho | Gos/ho - 512 health, 0.00 damage | |  |

## HTML parser

Write a program that extracts a title of a HTML file and all the content in its body. When you do that print the result in the following format:

**"Title: {extracted title}"**

**"Content: {extracted content}"**

The content should be a single string. There might be different tags inside of the body, which you must ignore. You extract only the text without the tags. The input will be on a single line. Example:

**"<html>\n<head><title>News</title></head>\n<body><p><a href="https://softuni.bg">Telerik\nAcademy</a>aims to provide free real-world practical\ntraining for young people who want to turn into\nskillful .NET software engineers.</p></body>\n</html>"**

Here the title is **"News"** and the content is **"Telerik Academy aims to provide free real-world practical training for young people who want to turn into skillful .NET software engineers."**

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comment** |
| <html>\n<head><title>Some title</title></head>\n<body>Here<p>is some</p>content<a href="www.somesite.com">\nclick</body>\n</html> | Title: Some title  Content: Here is some content click | We take the title and ignore all the tags to get the content |