Mini Exam: Regular Expressions

You can check your solutions here: https://judge.softuni.bg/Contests/3178/Additional-Exercises.

1. Treasures

Write a program that receives input of a string representing what you found in a hidden cave during your off-road track. There are always by one of two types of treasures: **precious metals** (gold, silver…) and **gems**. You will find by one piece of each type in the cave, but you need to extraxt them from the mess of garbage there with the following patterns:

- **@metal|**
- **#gem***

Print a line with the treasures in the following format:

"Found hidden {metal} and {gem} in the cave."

Example

Input	Output
rock grass @gold| dust #sapphire* sticks	Found hidden gold and sapphire in the cave.
Stones @silver| bones #ruby*	Found hidden silver and ruby in the cave.

2. Running Gear

Write a program that keeps track of bought running gear and the calculates the total cost of all items. You will receive some lines of input until the line "Run!". If a line is valid it will be in the following format:

"<>{gear name}<>{quantity}--{price}"

The price can be floating point number or whole number. Store the names of the gear and the total price. At the end print the each bought item on separate line in the format:

"Gear bought:

{1st name}

{2nd name}

…"

And on the last line print the following: **"Total cost: {spend money}"** formatted to the second decimal point.

Examples

Input	Output	Comment
<>Shoes<>1--349.99 <>Balaclava<>4-- <>Shorts<>5--185 Run!	Gear bought: Shoes Shorts Total cost: 1274.99	Only the Shoes and the Shorts are valid, for each of them we multiply the price by the quantity and print the result

3. Championship

Write a program that retrieves information about a road racing championship. On the **first line** you will get a **list of participants separated by ", "**. On the next few lines until you receive a line **"end of race"** you will get data lines containing some **alphanumeric characters**. In between them you could have some **extra characters which you should ignore**. For example: **"P!32e%t7e#32r$235@!6"**. The **letters are the name** of the person and the **sum of the**

digits is the distance he ran. So here we have **Peter** who ran **33 km**. Store the information about the person only **if the list of racers contains the name of the person**. If you receive the **same person more than once just add the distance to his old distance**. At the end **print the top 3 racers** ordered by **distance in descending** in the format:

"1st place: {first racer}

2nd place: {second racer}

3rd place: {third racer}"

Examples

Input	Output	Comment
Marian, Peter, Bill, Tom M4a@55ri%6a6!68n!!@ R1@!3a$y4456@ B5@i@#123ll M@a54r$i6an# 7P%et^#e5346r T$o553m&6 end of race	1st place: Marian 2nd place: Peter 3rd place: Tom	On the 3rd input line we have Ray. He is not in the list, so we do not count his result. The other ones are valid. George has total of 55 km, Peter has 25 and Tom has 19. We do not print Bill because he is on 4th place.