# JS Fundamentals Mid Exam Preparation

## Problem 1 - Counter-Strike

**Link:** <https://judge.softuni.org/Contests/Practice/Index/2305#0>

Write a program that **keeps track of every won** battle against an **enemy**. You will receive **initial energy**. Afterward, you will start receiving the **distance** you need **to reach an enemy** until the **"End of battle"** command is given, or you **run out of energy**.

The **energy** you need for reaching an enemy is **equal to the distance you receive**. Each time you reach an enemy, you **win** a battle, and your **energy is reduced**. Otherwise, if you don't have **enough energy** to reach an enemy, **end the program** and **print**: **"Not enough energy! Game ends with {count} won battles and {energy} energy"**.

Every **third won battle** increases **your energy with the value of your current count of won battles**.

Upon receiving the **"End of battle"** command**,** print the **count of won battles** in the following format:

**"Won battles: {count}. Energy left: {energy}"**

**Input / Constraints**

* On the **first line,** you will receive **initial energy** – an **integer [1-10000]**.
* On the **following lines,** you will be receiving the **distance** of an enemy – an **integer** **[1-10000].**

**Output**

* The description contains the proper output messages for each case and the format they should be printed.

**Examples**

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| (["100",  "10",  "10",  "10",  "1",  "2",  "3",  "73",  "10"]) | Not enough energy! Game ends with 7 won battles and 0 energy | The initial energy is 100. The first distance is 10, so we subtract 10 from 100, and we consider this a **won** battle. We are left with 90 energy. Next distance – 10, and 80 energy left.  Next distance – 10, 3 won battles and 70 energy, but since we have 3 won battles, we increase the energy with the current count of won battles, in this case – **3, and it becomes 73**.  The last distance we receive – **10** is unreachable since we have **0** energy, so we print the appropriate message, and the program ends. |
| (["200",  "54",  "14",  "28",  "13",  "End of battle"]) | Won battles: 4. Energy left: 94 |  |

## Problem 2 - Treasure Hunt

**Link:** <https://judge.softuni.org/Contests/Practice/Index/1773#1>

*The pirates need to carry a treasure chest safely back to the ship, looting along the way.*

Create a program that **manages** the **state** of the **treasure chest** along the way. On the **first line,** you will receive the **initial loot** of the treasure chest, which is a **string** of **items** separated by a **"|"**.

**"{loot1}|{loot2}|{loot3} … {lootn}"**

The following lines represent commands **until** **"Yohoho!"** which ends the treasure hunt:

* **"Loot {item1} {item2}…{itemn}"**:
  + Pick up treasure loot along the way. Insert the items at the **beginning** of the chest.
  + If an item is **already** contained, **don't** insert it.
* **"Drop {index}"**:
  + **Remove** the loot at the given **position** and **add** it at the **end** of the treasure chest.
  + If the index is **invalid,** skip the command.
* **"Steal {count}"**:
  + Someone steals the **last count** of loot items. If there are **fewer items** than the given count, **remove as many** as there are.
  + Print the stolen items separated by **", "**:

**"{item1}, {item2}, {item3} … {itemn}"**

In the end, output the **average treasure gain,** which is the **sum** of all treasure items' **length** divided by the **count** of all items inside the chest **formatted** to the **second decimal** point:

**"Average treasure gain: {averageGain} pirate credits."**

If the chest is **empty,** print the following message:

**"Failed treasure hunt."**

### Input

* On the **1st line,** you are going to receive the **initial treasure chest** (**loot** separated by **"|"**).
* On the following **lines**, until **"Yohoho!"**, you will be receiving commands.

### Output

* Print the output in the **format** **described** **above**.

### Constraints

* The **loot items** will be strings containing any ASCII code.
* The **indexes** will be integers in the range [**-200**…**200**]
* The **count** will be an integer in the range [**1**….**100**]

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| (["Gold|Silver|Bronze|Medallion|Cup",  "Loot Wood Gold Coins",  "Loot Silver Pistol",  "Drop 3",  "Steal 3",  "Yohoho!"]) | Medallion, Cup, Gold  Average treasure gain: 5.40 pirate credits. |
| **Comments** | |
| The first command **"Loot Wood Gold Coins"** adds **Wood** and **Coins** to the chest but **omits** Gold since it is already contained. The chest now has the following items:  **Coins Wood Gold Silver Bronze Medallion Cup**  The **second** command adds **only Pistol** to the chest  The **third** command **"Drop 3"** removes the **Gold** from the chest, but immediately adds it at the **end**:  **Pistol Coins Wood Silver Bronze Medallion Cup Gold**  The **fourth** command **"Steal 3"** removes the **last 3** items **Medallion**, **Cup**, **Gold** from the chest and prints them.  In the end calculate the average treasure gain which is the sum of all items length Pistol(**6**) + Coins(**5**) + Wood(**4**) + Silver(**6**) + Bronze(**6**) = **27** and **divide** it by the count 27 / 5 = **5.4** and format it to the **second decimal** point. | |
| **Input** | **Output** |
| (["Diamonds|Silver|Shotgun|Gold",  "Loot Silver Medals Coal",  "Drop -1",  "Drop 1",  "Steal 6",  "Yohoho!"]) | Coal, Diamonds, Silver, Shotgun, Gold, Medals  Failed treasure hunt. |

## Problem 3 - Inventory

**Link:** <https://judge.softuni.org/Contests/Practice/Index/2028#2>

*As a young traveler, you gather items and craft new items.*

### Input / Constraints

You will receive a journal with some collecting items, separated with a comma and a space (**", "**). After that, until receiving "Craft!" you will be receiving different commands split by **" - "**:

* "Collect - {item}" - you should add the given item to your inventory. If the item already **exists**, you should **skip** this line.
* "Drop - {item}" - you should remove the item from your inventory **if it exists**.
* "Combine Items - {old\_item}:{new\_item}" - you should check if the **old item exists**. If so, **add** the new item **after** the old one. Otherwise, **ignore** the command.
* "Renew – {item}" – if the given item exists, you should change its position and **put it last** in your inventory.

### Output

After receiving "Craft!" print the items in your inventory, separated by **", "**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| [  'Iron, Wood, Sword',  'Collect - Gold',  'Drop - Wood',  'Craft!'  ] | Iron, Sword, Gold |
| [  'Iron, Sword',  'Drop - Bronze',  'Combine Items - Sword:Bow',  'Renew - Iron',  'Craft!'  ] | Sword, Bow, Iron |