# Programming Fundamentals with Python: Exam Preparation

## Counter-Strike

**[Link to Judge](https://judge.softuni.org/Contests/Practice/Index/2305" \l "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 to reach 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 receive an enemy's distance – 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 |  |

## Array Modifier

**[Link to Judge](https://judge.softuni.org/Contests/Practice/Index/2474" \l "1)**

You are given **an array with integers**. Write a program to **modify the elements** after **receiving the following commands**:

* "swap {index1} {index2}" takes **two elements** and **swaps their places**.
* "multiply {index1} {index2}" takes **element at the 1st index** and **multiply** **it** **with the element at 2nd index**. **Save the product at the 1st index.**
* "decrease" **decreases** **all elements** in the array **with 1**.

### Input

On the **first input line,** you will be given **the initial array values** separated by a single space.

On the **next lines,** you will receive commands **until** you receive the **command "end"**. The **commands are** as follows:

* "swap {index1} {index2}"
* "multiply {index1} {index2}"
* "decrease"

### Output

**The output** should be printed on the console and consist of **elements** **of the** **modified array** – **separated by a comma and a single space** "**,** ".

### Constraints

* **Elements of the array** will be **integer numbers** in the range **[-231...231].**
* **The count of the array elements** will be in the range **[2...100].**
* **Indexes** **will always be in the range of the array.**

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 23 -2 321 87 42 90 -123  swap 1 3  swap 3 6  swap 1 0  multiply 1 2  multiply 2 1  decrease  end | 86, 7382, 2369942, -124, 41, 89, -3 | 23 -2 321 87 42 90 -123 – initial values  swap 1(-2) and 3(87) ▼  23 87 321 -2 42 90 -123  swap 3(-2) and 6(-123) ▼  23 87 321 -123 42 90 -2  swap 1(87) and 0(23) ▼  87 23 321 -123 42 90 -2  multiply 1(23) 2(321) = 7383 ▼  87 7383 321 -123 42 290 -2  multiply 2(321) 1(7383) = 2369943 ▼  87 7383 2369943 -123 42 90 -2  decrease – all - 1 ▼  86 7383 2369942 -124 41 89 -3 |
| 1 2 3 4  swap 0 1  swap 1 2  swap 2 3  multiply 1 2  decrease  end | 1, 11, 3, 0 |  |

## Inventory

**[Link to Judge](https://judge.softuni.org/Contests/Practice/Index/2028" \l "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 |