# Mid Exam Preparation – 17 October 2022

## Guinea Pig

**Link:** [**https://judge.softuni.org/Contests/Practice/Index/2031#0**](https://judge.softuni.org/Contests/Practice/Index/2031#0)

*Merry has a guinea pig named Puppy, that she loves very much. Every month she goes to the nearest pet store and buys him everything he needs – food, hay, and cover.*

On the **first three lines**, you will receive **the quantity of food**, **hay**, and **cover**, which Merry buys for a **month (30 days)**. On the **fourth line**, you will receive the **guinea pig's weight**.

**Every day** Puppy eats **300 gr of food**. **Every** **second** day Merry **first feeds the pet**, then gives it a **certain amount of hay** **equal to** **5%** of the rest of the **food**. On **every** **third** day, Merry puts Puppy **cover** with **a quantity of** **1/3** of its **weight**.

**Calculate** whether the quantity of **food, hay, and cover**, will be enough for a **month**.

**If Merry runs out of food, hay, or cover, stop the program!**

## Input

* **On the first line – quantity food in kilograms** - afloating-point number in the range **[0.0 – 10000.0].**
* **On the second line – quantity hay in kilograms** - afloating-point number in the range **[0.0 – 10000.0].**
* **On the third line – quantity cover in kilograms** - afloating-point number in the range **[0.0 – 10000.0].**
* **On the fourth line – guinea's weight in kilograms** - afloating-point number in the range **[0.0 – 10000.0].**

## Output

* If the food, the hay, and the cover are enough, print:
  + **"Everything is fine! Puppy is happy! Food: {excessFood}, Hay: {excessHay}, Cover: {excessCover}."**
* If one of the things is not enough, print:
  + **"Merry must go to the pet store!"**

**The output values must be formatted to the second decimal place!**

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10  5  5.2  1 | Everything is fine! Puppy is happy! Food: 1.00, Hay: 1.10, Cover: 1.87. |
| You receive food – **10000**, hay – **5000**, cover – **5200**, weight – **1000** (in grams).  On the first day, Merry gives Puppy 300gr food – 9700gr food left.  On the second day, the food left is **9400gr**, so the needed hay is **9400 \* 5% = 470**,and thehay left is **4530.**  On the third day, the cover left is **4866.67,** and the food left is **9100**,and so on.  On the last day, Merry has: food – 1.00, hay – 1.10, and cover – 1.87. | |
| 1  1.5  3  1.5 | Merry must go to the pet store! |
| 9  5  5.2  1 | Merry must go to the pet store! |

## Array Modifier

**Link:** [**https://judge.softuni.org/Contests/Practice/Index/2474#1**](https://judge.softuni.org/Contests/Practice/Index/2474#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 **swap their places**.
* "multiply {index1} {index2}" takes **the element at the 1st index** and **multiplies** **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].**
* **Count of the array elements** will be in the range **[2...100].**
* **Indexes** **will be always 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:** [**https://judge.softuni.org/Contests/Practice/Index/2028#2**](https://judge.softuni.org/Contests/Practice/Index/2028#2)

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 |
| **Input** | **Output** |
| Iron, Sword  Drop - Bronze  Combine Items - Sword:Bow  Renew - Iron  Craft! | Sword, Bow, Iron |