Problem 1 – Bonus Scoring System

Create a program that calculates **bonus points** for each **student** enrolled in a course. On the **first** line, you are going to receive **the number of the students**. On the **second** line, you will receive **the total number of lectures** in the course. The course has **an additional bonus**, which you will receive **on the third line**. On the following lines, you will be receiving the **count of attendances for each student**.

The bonus is calculated with the following **formula**:

`{total bonus} = {student attendances} / {course lectures} * (5 + {additional bonus})`

Find the student with the **maximum bonus** and print them, along with **his attendances,** in the following format:

`"Max Bonus: {max bonus points}."`

`"The student has attended {student attendances} lectures."`

Round the bonus points at the end to **the nearest larger number**.

Input / Constrains

- On the **first line,** you are going to receive the **number of the students** – an integer in the range [**0…50**]
- On the **second line,** you will receive the **number of the lectures** – an integer number in the range [**0...50**].
- On the **third line**, you will receive **the additional bonus** – an integer number in the range [**0…100**].
- **On the following lines**, you will be receiving the **attendance of each student**.
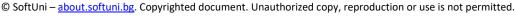- There will **never** be **students with equal bonuses**.

Output

- Print the **maximum bonus points** and the **attendances** of the given student, **rounded** to the nearest **larger** number, scored by a student in this course in the format described above.

Examples

Input	Output
5 25 30 12 19 24 16 20	Max Bonus: 34. The student has attended 24 lectures.

First, we receive the **number of students** enrolled in the course – **5**. The total count of the lectures is **25,** and the additional bonus is **30**. Then we calculate the bonus of the student with **12** attendances, which is **16.8**. We continue calculating **each of the student's bonuses**. The one **with 24 attendances** has the **highest bonus – 33.6 (34 rounded)**, so we print the appropriate message on the console.

10 30 14 8 23 27 28 15 17 25 26 5 18	Max Bonus: 18. The student has attended 28 lectures.

Problem 2 – Array Modifier

A problem for exam preparation for the "C# Fundamentals" course @ SoftUni
Submit your solutions in the SoftUni Judge system here

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 **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 follow:

- **"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	

Problem 3 – Inventory

A problem for exam preparation for the "C# Fundamentals" course @ SoftUni
Submit your solutions in the SoftUni Judge system here

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

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