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.
Comments	

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.

JS Input / Output

Input	Output
['5', '25', '30', '12', '19', '24', '16', '20']	Max Bonus: 34. The student has attended 24 lectures.
Comments	
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. Mu Online

Problem for exam preparation for the Programming Fundamentals Course @SoftUni.
Submit your solutions in the SoftUni judge system at https://judge.softuni.org/Contests/Practice/Index/2028#1.

You have **initial health 100 and initial bitcoins 0**. You will be given **a string representing the dungeon's rooms**. Each room is separated with **'|'** (vertical bar): **"room1|room2|room3…"**

Each room contains **a command** and a **number**, separated by space. The command can be:

- **"potion"**
  - You are healed with the number in the second part. But your health **cannot exceed** your **initial health (100)**.
  - First print: **"You healed for {amount} hp."**
  - After that, print your current health: **"Current health: {health} hp."**
- **"chest"**
  - You've found some bitcoins, the number in the second part.
  - Print: **"You found {amount} bitcoins."**
- In **any other case,** you are **facing a monster**, which you will **fight**. The **second part of the room** contains the **attack** of the monster. You should remove the monster's attack from your health.
  - If you are not dead (health <= 0), you've slain the monster, and you should print: **"You slayed {monster}."**
  - If you've died, print **"You died! Killed by {monster}."** and your quest is over. Print the best room you've manage to reach: **"Best room: {room}"**

If you managed to **go through all the rooms** in the dungeon, print on the **following three lines**:

"You've made it!"

"Bitcoins: {bitcoins}"

"Health: {health}"

Input / Constraints

You receive a **string** representing the dungeon's rooms, separated with **'|'** (vertical bar): **"room1|room2|room3…"**.

Output

Print the corresponding messages described above.

Examples

Input	Output

rat 10|bat 20|potion 10|rat 10|chest 100|boss 70|chest 1000	You slayed rat. You slayed bat. You healed for 10 hp. Current health: 80 hp. You slayed rat. You found 100 bitcoins. You died! Killed by boss. Best room: 6
Input	**Output**
cat 10|potion 30|orc 10|chest 10|snake 25|chest 110	You slayed cat. You healed for 10 hp. Current health: 100 hp. You slayed orc. You found 10 bitcoins. You slayed snake. You found 110 bitcoins. You've made it! Bitcoins: 120 Health: 65

JS Input / Output

Input	Output
"rat 10|bat 20|potion 10|rat 10|chest 100|boss 70|chest 1000"	You slayed rat. You slayed bat. You healed for 10 hp. Current health: 80 hp. You slayed rat. You found 100 bitcoins. You died! Killed by boss. Best room: 6
Input	**Output**
"cat 10|potion 30|orc 10|chest 10|snake 25|chest 110"	You slayed cat. You healed йfor 10 hp. Current health: 100 hp. You slayed orc.

	You found 10 bitcoins. You slayed snake. You found 110 bitcoins. You've made it! Bitcoins: 120 Health: 65

Problem 3. Inventory

Problem for exam preparation for the Programming Fundamentals Course @SoftUni.
Submit your solutions in the SoftUni judge system at 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

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

JS Input / Output

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