

Python Advanced: Exam Preparation

1. Collecting Eggs

Link to Judge: <https://judge.softuni.org/Contests/Practice/Index/3515#0>

Old MacDonald wants to fill some boxes with eggs. But he has a big farm, and he will need some help.

On the first line, you will receive a sequence of **numbers**, each representing an **egg with its size**. On the second line, you will receive another sequence of **numbers**, each representing a **piece of paper with its size**.

You should **take the first egg** and wrap it with the **last piece of paper**. Then, try to put it in a **box with a size of 50**. Each **wrapped-in-a-paper egg** fills **one box** if it fits in it. Your task is to check **whether you have filled at least one box**.

You should comply with the following conditions:

- If the egg is not fresh anymore (**its size is less than or equal to 0**), you need to **remove it from the sequence before** it is wrapped with a piece of paper.
- If the **sum** of the egg's size and the paper's size is **less than or equal to the box's size (50)**, **put the wrapped egg in the box** and **remove both** from the sequences.
 - Otherwise, you **cannot fill a box**, so **remove both** the egg and the paper from the sequences **without putting them in a box**.
- During your work, you noticed that Old MacDonald is superstitious. If the **size of an egg is 13** it brings bad luck to him. You should **remove this egg** from the sequence **before** it is wrapped with a piece of paper.
 - Furthermore, each time you take an egg with a size of 13, it will be best to **swap the first and last pieces of paper positions** to bring the good luck back to Old MacDonald.
 - Note: There will be **NO case** where there will be just **one piece of paper** left.

For more clarification see the examples below.

Input

- In the **first line**, you will be given a **sequence of eggs with their sizes - integers** separated by comma and space ", " in the range **[-100, 100]**
- In the **second line**, you will be given a **sequence of pieces of paper with their sizes - integers** separated by comma and space ", " in the range **[1, 100]**

Output

- On the first line:
 - If you have **at least one box filled**, print:
 - **"Great! You filled {total count} boxes."**
 - If you **couldn't fill any boxes**, print:
 - **"Sorry! You couldn't fill any boxes!"**
- On the following lines, print the eggs left or pieces of paper left **if there are any**:
 - **Eggs left: {left eggs joined by ", "}**
 - **Pieces of paper left: {left pieces of paper joined by ", "}**

Constraints

- You will always have **at least one egg** and **at least one piece of paper**.

Examples

Input	Output
20, 13, -7, 7 10, 5, 20, 15, 7, 9	Great! You filled 2 boxes. Pieces of paper left: 7, 5, 20, 15
Comment	
1) The first egg (20) is wrapped with the last piece of paper (9). We put them in a box and remove them from the sequences. 2) The second egg (13) brings back luck so it's removed. Then the first piece of paper (10) is switched with the last piece of paper (7). 3) The third egg (-7) is not fresh, so we remove it. 4) The fourth egg (7) is wrapped with the last piece of paper (10). We put them in a box and remove them from the sequences. Remove them both. 5) We successfully filled 2 boxes.	
Input	Output
2, 4, 7, 8, 0 5, 6, 2	Great! You filled 3 boxes. Eggs left: 8, 0
Input	Output
12, 23 28, 40	Sorry! You couldn't fill any boxes!

2. CRUD

Link to Judge: <https://judge.softuni.org/Contests/Practice/Index/3534#1>

*The abbreviation CRUD expands to Create, Read, Update and Delete.
These are the four fundamental operations in a database.*

In the beginning, you will be given a **matrix** with **6 rows and 6 columns** representing a table with information. It consists of:

- **Letters** - on one or many positions in the table
- **Numbers** - on one or many positions in the table
- **Empty positions** - marked with "."

Next, you will receive your **first position** on the table in the format "**{row}, {column}**)"

On the following lines, until you receive "**Stop**" you will be receiving **commands** in the format:

- "**Create, {direction}, {value}**"
 - The direction could be "**up**", "**down**", "**left**" or "**right**"
 - If you **step in an empty position**, **create** the given **value** on that position. E.g., if the given value is "**A**", and the position is empty (".") - change it to "**A**"
 - If the position is **NOT empty**, do **NOT create** a value on that position

- **"Update, {direction}, {value}"**
 - The direction could be "up", "down", "left" or "right"
 - If you **step on a letter or number, update** the position with the given **value**. E.g., if the given value is "h", and the position's value is "12" - change it to "h"
 - If the position is **empty**, do **NOT update** the value on that position
- **"Delete, {direction}"**
 - The direction could be "up", "down", "left" or "right"
 - If you **step on a letter or number, delete** it, and **empty** the position. E.g., if the given position's value is "h" - change it to "."
 - If the position is **already empty**, do **NOT delete** it
- **"Read, {direction}"**
 - The direction could be "up", "down", "left" or "right"
 - If you **step on a letter or number, print it** on the console
 - If the position is **empty**, do **NOT read** it

You can make **only ONE move at a time in the given direction** for each command given.

In the end, **print the final matrix**.

Input

- **On the first 6 lines - a matrix** with positions **separated by a single space**
 - **Letters** are in the range [a-zA-Z]
 - **Numbers** are in the range [-100, 100]
- On the next line - your **first position** in the format: "{row}, {column}"
- On the following lines until you receive the command "Stop" - commands in the format shown above

Output

- In the end, **print the final matrix**, each row on a new line, each position separated by a single space.

Constraints

- You will **always** receive **valid coordinates**
- You will **always** receive **directions in the range of the table**
- You will **always** receive **letters or numbers**

Examples

Input	Output	Comment
. 6 G . S . t S . . 10 . . .	t 6 G r e a t .	Start from the position (1, 1). 1) The first command is "Create", the direction is "down" and the value is "r". Create the value "r" on the empty position (2, 1).

<pre> . 95 . . 8 . . . P . . . (1, 1) Create, down, r Update, right, e Create, right, a Read, right Delete, right Stop </pre>	<pre> . . 10 95 . . 8 . . . P . . . </pre>	<p>2) The next command is "Update", the direction is "right" and the value is "e". We change the old value "S" on position (2, 2) with the value "e".</p> <p>3) The next command is "Create", the direction is "right" and the value is "a". Create the value "a" on the empty position (2, 3).</p> <p>4) The next command is "Read", the direction is "right". Print the value "t" on position (2, 4).</p> <p>5) The next command is "Delete", the direction is "right". Delete the value "S" on position (2, 5).</p> <p>6) Receive the command "Stop", print the final matrix, and end the program.</p>
<pre> 6 T . D . O . . 10 A . . . 95 . 80 5 . . . P . t . (2, 3) Create, down, o Delete, right Read, up Create, left, 20 Update, up, P Stop </pre>	<pre> 6 T . D . O . . 10 A . . . 95 . 80 5 . . . P . t . </pre>	
<pre> H 8 70 i t . . . B . 50 . 16 . C t . . . 25 (0, 0) Read, right Read, down Read, left Delete, down </pre>	<pre> 8 i 70 H 8 70 i 10 . . B . 50 . 16 . C t . . . 25 </pre>	

Create, right, 10		
Read, left		
Stop		

3. Song Creator

Link to Judge: <https://judge.softuni.org/Contests/Practice/Index/3534#2>

Create a function called `add_songs()`.

It receives **one or many tuples**. Each tuple consists of exactly two elements - the **song's title** in the **first position** and a **list** in the **second position**. The list can consist of **one, many, or no strings** - each representing a **line** of the lyrics of the song.

The function **collects the information** and **concatenates the lyrics for each song** (each string on a **different line**). If you are given the **same song more than once**, **add the additional lyrics** (if ones are given) to the lyrics of the song.

In the end, it should **return a string for each song with its lyrics** in the format:

```
"- {song_title}"
"{first_line_of_lyrics}"
"{second_line_of_lyrics}"
...
"{nth_line_of_lyrics}"
```

If there are **no lyrics** given for a song, return **just its title in the format shown above**.

For more clarification, see the examples below.

Input

- There will be **no input**, just tuples passed to your function.

Output

- Return** the desired result as described above.

Constraints:

- You will always have a **song's name on the first position of the tuple**.

Examples

Input	Output
<pre>print(add_songs(("Bohemian Rhapsody", []), ("Just in Time", ["Just in time, I found you just in time", "Before you came, my time was running low", "I was lost, the losing dice were tossed", "My bridges all were crossed, nowhere to go"])))</pre>	<pre>- Bohemian Rhapsody - Just in Time Just in time, I found you just in time Before you came, my time was running low I was lost, the losing dice were tossed My bridges all were crossed, nowhere to go</pre>

<pre> "I was lost, the losing dice were tossed", "My bridges all were crossed, nowhere to go"]))) </pre>	
Input	Output
<pre> print(add_songs(("Beat It", []), ("Beat It", ["Just beat it (beat it), beat it (beat it)", "No one wants to be defeated"]), ("Beat It", []), ("Beat It", ["Showin' how funky and strong is your fight", "It doesn't matter who's wrong or right"])),)) </pre>	<pre> - Beat It Just beat it (beat it), beat it (beat it) No one wants to be defeated Showin' how funky and strong is your fight It doesn't matter who's wrong or right </pre>
Input	Output
<pre> print(add_songs(("Love of my life", ["Love of my life, you've hurt me", "You've broken my heart, and now you leave me", "Love of my life, can't you see?", "Bring it back, bring it back"]), ("Beat It", []), ("Love of my life", ["Don't take it away from me", "Because you don't know", "What it means to me"]), ("Dream On", ["Every time that I look in the mirror"])),)) </pre>	<pre> - Love of my life Love of my life, you've hurt me You've broken my heart, and now you leave me Love of my life, can't you see? Bring it back, bring it back Don't take it away from me Because you don't know What it means to me - Beat It - Dream On Every time that I look in the mirror </pre>