Exercise: Arrays

Problems for exercise and homework for the "JS Fundamentals" Course @ SoftUni.
Submit your solutions in the SoftUni judge system at: https://judge.softuni.org/Contests/1256

1. Add and Subtract

Write a function, which changes the **value** of **odd** and **even** numbers in an array of numbers.

- If the number is **even** - **add** to its value its **index** position
- If the number is **odd** - **subtract** to its value its index position

Output

On the first line print the **newly modified array**, on the second line print the **sum** of numbers from the **original array**, on the third line print the sum of numbers from the **modified array**.

Examples

Input	Output

`[5, 15, 23, 56, 35]`	`[ 5, 14, 21, 59, 31 ]`
	`134`
	`130`

`[-5, 11, 3, 0, 2]`	`[ -5, 10, 1, 3, 6 ]`
	`11`
	`15`

2. Common Elements

Write a function, which prints common elements in **two string arrays**. Print all matches on **separate** lines. Compare the **first array** with the **second array**.

Examples

Input	Output
`['Hey', 'hello', 2, 4, 'Peter', 'e'],` `['Petar', 10, 'hey', 4, 'hello', '2']`	`hello` `4`
`['S', 'o', 'f', 't', 'U', 'n', 'i', ' '],` `['s', 'o', 'c', 'i', 'a', 'l']`	`o` `i`

3. Merge Arrays

Write a function, which receives **two string arrays** and **merges** them into a **third** array.

- If the **index** of the element is **even**, add into the third array the **sum** of **both elements** at that index

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

- If the **index** of the element is **odd**, add the **concatenation** of both elements at that index

Input

As **input,** you will receive **two string arrays (**with equal length**).**

Output

As **output,** you should print the resulting **third array**, each element separated by **" - "**.

Examples

Input	Output
['5', '15', '23', '56', '35'], ['17', '22', '87', '36', '11']	22 - 1522 - 110 - 5636 - 46
['13', '12312', '5', '77', '4'], ['22', '333', '5', '122', '44']	35 - 12312333 - 10 - 77122 - 48

4. Array Rotation

Write a function that receives an **array** and the **number of rotations** you have to perform.

Note: Depending on the number of rotations, the first element goes to the end.

Output

Print the resulting array elements separated by a single space.

Examples

Input	Output
[51, 47, 32, 61, 21], 2	32 61 21 51 47
[32, 21, 61, 1], 4	32 21 61 1
[2, 4, 15, 31], 5	4 15 31 2

5. Max Number

Write a function to find all the **top** integers in an array. A top integer is an integer, which is **bigger** than all the elements to its **right**.

Output

Print **all** top integers on the console, **separated** by a single space.

Examples

Input	Output
[1, 4, 3, 2]	4 3 2

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

[14, 24, 3, 19, 15, 17]	24 19 17
[41, 41, 34, 20]	41 34 20
[27, 19, 42, 2, 13, 45, 48]	48

6. Equal Sums

Write a function that determines if there exists an element in the array of numbers such that the sum of the elements on its **left** is **equal** to the sum of the elements on its **right**.

If there are **NO elements** to the **left/right**, their **sum** is **0**.

Print the **index** that satisfies the required condition or **"no"** if there is no such index.

Examples

Input	Output	Comments
[1, 2, 3, 3]	2	At a[2] -> left sum = 3, right sum = 3 a[0] + a[1] = a[3]
[1, 2]	no	At a[0] -> left sum = 0, right sum = 2 At a[1] -> left sum = 1, right sum = 0 No such index exists
[1]	0	At a[0] -> left sum = 0, right sum = 0
[1, 2, 3]	no	No such index exists
[10, 5, 5, 99, 3, 4, 2, 5, 1, 1, 4]	3	At a[3] -> left sum = 20, right sum = 20 a[0] + a[1] + a[2] = a[4] + a[5] + a[6] + a[7] + a[8] + a[9] + a[10]

7. Max Sequence of Equal Elements

Write a function that finds the **longest sequence** of **equal elements** in an array of numbers.

If several longest sequences exist, print the **leftmost** one.

Examples

Input	Output
[2, 1, 1, 2, 3, 3, 2, 2, 2, 1]	2 2 2
[1, 1, 1, 2, 3, 1, 3, 3]	1 1 1
[4, 4, 4, 4]	4 4 4 4
[0, 1, 1, 5, 2, 2, 6, 3, 3]	1 1

8. Magic Sum

Write a function, which prints all **unique** pairs in an **array of integers** whose **sum** is **equal** to a given number.

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.
Follow us:

Examples

Input	Output
[1, 7, 6, 2, 19, 23], 8	1 7 6 2
[14, 20, 60, 13, 7, 19, 8], 27	14 13 20 7 19 8
[1, 2, 3, 4, 5, 6], 6	1 5 2 4

9. *Dungeonest Dark

As a young adventurer, you seek gold and glory in the darkest dungeons there are.

You have **initial health 100** and **initial coins 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 - an **item** or a **monster**; and a number. They are separated by a single space.

(**"item/monster number"**).

- If the first part is **"potion"**:
  - You are healed with the number in the second part. However, your health **cannot exceed** your **initial health (100)**.
  - Print: **`You healed for {healing-number} hp.`**
  - After that, print your current health: **`Current health: {number} hp.`**
- If the first part is **"chest"**:
  - You have found some coins, the number in the second part.
  - Print: **`You found {coins} coins.`**
- In any other case, you are facing a monster, you are going to fight.

  The second part of the room contains the attack of the monster, and the first the monster's name. You should remove the monster's attack from your health.
  - If you are not dead (**health > 0**) you have slain the monster, and you should print:

    `You slayed {monster-name}.`
  - If you have died, print: **`You died! Killed by {monster-name}.`** and your quest is over. Print the best room you've to manage to reach: **`Best room: {room}`**.
- If you managed to go through all the rooms in the dungeon, print on the next three lines:

  "You've made it!"

  "Coins: {coins}"

  "Health: {health}"

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

Input

You receive an array with one element- 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 coins. You died! Killed by boss. Best room: 6
["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 coins. You slayed snake. You found 110 coins. You've made it! Coins: 120 Health: 65

...! a game where every hero wins the day with shiny armor and a smile...

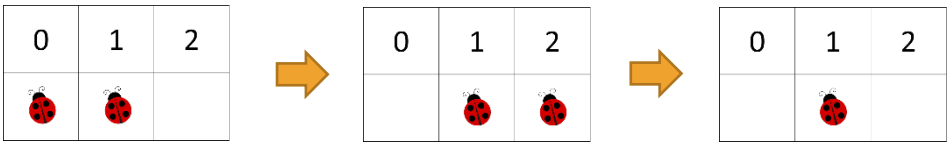
10. * Ladybugs

You are **given a field size** and the **indexes of ladybugs** inside the field. A **ladybug changes its position** either to its **left or to its right by a given fly length**. A **command to a ladybug** looks like this: "**`0 right 1`**". This means that the little insect placed on index 0 should fly one index to its right. If the ladybug **lands on a fellow ladybug**, it **continues to fly** in the same direction **by the same fly length**. If the ladybug **flies out of the field, it is gone**.

For example, imagine you are given a field with size 3 and ladybugs on indexes 0 and 1. If the ladybug on index 0 needs to fly to its right by the length of 1 (0 right 1) it will attempt to land on index 1 but as there is another ladybug there it will continue further to the right by an additional length of 1, landing on index 2. After that, if the same

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.
Follow us:

ladybug needs to fly to its right by the length of 1 (2 right 1), it will land somewhere outside of the field, so it flies away:



If you are given a ladybug index that does not have a ladybug there, nothing happens. If you are given a ladybug index that is outside the field, nothing happens.

Your job is to create the program, simulating the ladybugs flying around doing nothing. In the end, **print all cells in the field separated by blank spaces**. For each cell that has a ladybug on it print '**1**' and for each empty cell print '**0**'. For the example above, the output should be '**0 1 0**'.

Input

- You will receive an **array of strings** and the first element is an **integer** – the **size** of the field
- The second element is a **string** containing the initial **indexes** of all ladybugs separated by a blank space. **The given indexes** may or may not be inside the field range
- The next elements in the **array** are commands in the format:
  "**{ladybug index} {direction} {fly length}**"

Output

- Print the **all cells within the field in format: `{cell} {cell} … {cell}`**
  - If a cell has a ladybug in it, print '**1**'
  - If a cell is empty, print '**0**'

Constrains

- The size of the field will be in the range [0 … 1000]
- The ladybug indexes will be in the range [-2,147,483,647 … 2,147,483,647]
- The number of commands will be in the range [0 … 100]
- The fly length will be in the range [-2,147,483,647 … 2,147,483,647]

Examples

Input	Output	Comments
[3, '0 1', '0 right 1', '2 right 1']	0 1 0	1 1 0 - Initial field 0 1 1 - field after "0 right 1" 0 1 0 - field after "2 right 1"

Input	Output

Input	Output

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.
Follow us:

[3, '0 1 2', '0 right 1', '1 right 1', '2 right 1']	0 0 0

[5, '3', '3 left 2', '1 left -2']	0 0 0 1 0

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.