少年圖靈計畫
Young turing Program

0_送分題 (Hello World)

(30分)

時間限制: 1 second
記憶體限制: 256 MB

前言

比賽開始了！

趕快驗證一下，

網路是否設定正確?

上傳競賽程式是否順利?

程式解答是否用 STDOUT 輸出？

都沒問題，30分就到手了！ 繼續 ... 衝！衝！衝！

題目敘述

請寫一個程式輸出Hello World!

輸入格式

本題無需輸入值

輸出格式

[A~Z][a~z]、空格，以及常用英文符號。

資料範圍

[A~Z][a~z]、空格，以及驚嘆號 “!”

測試範例

輸入範例 1

(無輸入值)

輸出範例 1

```
Hello World!
```

範例說明

輸入範例1, 無輸入值，簡單而快樂的輸出Hello World!

0_Hello World

(30 points)

Time Limit: 1 second
Memory Limit: 256MB

Introduction

YTP Contest has started!

Let’s verify everything first.

Is the internet setting correct?

Is the source code submission working well?

Do you use STDOUT output for program solutions?

Everything is ready! Go get 30 points now!! Go! Go! Go!

Statement

Please write a program to output Hello World!

Input Format

This problem requires no input.

Output Format

[A~Z][a~z], space, and common English punctuation.

Constraints

[A~Z][a~z], space, and exclamation mark "!".

Test Cases

Input 1

(no input)

Output 1

```
Hello World!
```

Illustrations

Input 1 has no input, simply output Hello World!

1_NasaLee vs Vegetable

(5分)

時間限制: 1 second
記憶體限制: 256 MB

題目敘述

在遙遠的東方世界中，有一個名為「NASA」的國度。NASA 是一個獨裁專制的國家，由領導人 Vegetable 統治，他在這個國家中扮演著如同國王一樣的角色。不僅全國人民都需要對他卑躬屈膝，舉國上下的大小事物也都需要經過他的決定。不過，由於要決定的事項實在太多，Vegetable 有時沒辦法一個人處理完，因此，他在國民中精挑細選出了一些優秀的人才協助他辦公，並稱作「佇」。由於這些佇在國家中扮演著重要的角色，因此，人民又習慣稱呼他們為「佇角」。

NasaLee 是生長於 NASA 的一名青年。從小開始，他就對著佇角們有著很大的憧憬，他的夢想便是希望能夠成為他們的一員，協助他所尊敬的 Vegetable 做事。也因此，他在成年的這一年便成為了見習佇角，開始了實習的生活。然而，在實習的期間，他卻發現現實跟他想的完全不一樣。佇角的工作量不僅大的不合理，還得隨時聽候 Vegetable 的差遣。最令他不可置信的是，他尊敬的 Vegetable 平時在國民眼中表現出一幅和藹的面孔，在工作方面，他對待佇角的態度卻十分惡劣，幾乎到了不把他們當人看的地步。感到失望的 NasaLee 於是決定起身反抗 Vegetable，他組織了一支起義軍隊，準備和 Vegetable 進行殊死決戰。

NasaLee 和 Vegetable 現在各自擁有一支軍隊，雙方的戰力值各自可以用一個正整數表示。NasaLee 的軍隊戰力值為 $NasaLee$，Vegetable 的軍隊戰力值則為 $Vegetable$。在最終決戰中，戰力值比較大的那一方將成為勝者。你，作為一位史學家，需要紀錄下這場戰役的勝者，讓後人能夠銘記這段歷史。

輸入格式

輸入有一行，每行包含兩個正整數 $NasaLee, Vegetable$。

輸出格式

輸出一個字串代表勝者的名字 "NasaLee"(不含括號) 或是 "Vegetable"(不含括號)，保證不會有戰力相等的情況。

資料範圍

- $1 \leq NasaLee, Vegetable \leq 10^5$

測試範例

輸入範例 1

```
7 4
```

輸出範例 1

```
NasaLee
```

輸入範例 2

```
5 12
```

輸出範例 2

```
Vegetable
```

1_NasaLee vs Vegetable

(5 points)

Time Limit: 1 second
Memory Limit: 256MB

Statement

In the distant Eastern world, there is a nation called "NASA." NASA is an autocratic country ruled by a leader named Vegetable, who plays a king-like role in the country. Not only do all the people have to bow to him, but all matters of the country, big or small, must be decided by him. However, due to the sheer number of decisions to be made, Vegetable sometimes cannot handle everything by himself. Therefore, he carefully selects outstanding talents from among the citizens to assist him in his office, calling them the "The Alliance," which is also known as "TA."

NasaLee is a young man born in NASA. From a young age, he has looked up to the TAs with great admiration, dreaming of becoming one of them and assisting the respected Vegetable in his work. Thus, in the year he became an adult, he became a trainee TA, starting his internship. However, during his internship, he discovered that reality was completely different from what he had imagined. The workload of a TA was not only unreasonably large, but they also had to be at Vegetable's beck and call at all times. What shocked him the most was that the Vegetable he respected, who usually showed a kindly face to the citizens, treated the TAs very poorly at work, almost to the point of dehumanizing them. Disappointed, NasaLee decided to rise up against Vegetable. He organized a rebel army, preparing for a decisive battle against Vegetable.

NasaLee and Vegetable each now have an army, and the combat power of each side can be represented by a positive integer. NasaLee's army's combat power is $NasaLee$, and Vegetable's army's combat power is $Vegetable$. In the final battle, the side with the higher combat power will be the winner. As a historian, you need to record the winner of this battle so that future generations can remember this historical event.

Input Format

The input consists of one line containing two positive integers $NasaLee$ and $Vegetable$.

Output Format

Output a string representing the name of the winner: "NasaLee" (without quotes) or "Vegetable" (without quotes). It is guaranteed that their combat powers would not be the same.

Constraints

- $1 \leq NasaLee, Vegetable \leq 10^5$

Test Cases

Input 1

```
7 4
```

Output 1

```
NasaLee
```

Input 2

```
5 12
```

Output 2

```
Vegetable
```

2_Vim vs Code

(10 分)

時間限制: 1 second

記憶體限制: 256 MB

題目敘述

你都用 Vim 還是 Vscode？

Vim 和 Vscode 之爭，不僅是文字編輯器之間的較量，更是傳統與前衛的對決、經典和新潮的碰撞。於是，程式設計師們形成了兩大教派，Vim 教派和 Vscode 教派，彼此都想證明自己擁護的才是編輯器的正統！

這天，Vscode 教徒有了一個邪惡的計畫。他們認為，Vim 區分一般模式和編輯模式的設計實在是太反人類了，怎麼會有人想用 HJKL 鍵當方向鍵來用？因此他們在 Vim 裡面植入了「Vim Suspiciously Compromised On Direction-keys Entered」插件，簡稱 VSCODE。被植入 VSCODE 插件的 Vim 編輯器會將一些字母誤認為方向鍵，導致使用者無法正常輸入，讓使用者能夠意識到 Vim 的設計多麼不合理。

具體來說，Vim 是一個網格狀的編輯區域，高 H 格寬 W 格、共有 $H \times W$ 格。編輯區域中每一格都可以顯示一個字元，一開始每一格都是空白。此外，編輯時會有一個「游標」決定輸入的位置，一開始在最左上角的格子。當使用者輸入一個字元時，游標所在的格子會填上該字元，且游標往右移動一格；若游標已經在最右邊一行，則游標移動到下一列的最左邊；若游標已經在最右下角，則游標回到最左上角。但因為被植入 VSCODE 插件的關係，編輯時如果輸入了以下四種字元，他們將會被當作方向鍵來處理而不修改編輯區域的內容：

- 輸入小寫字母 `h`，游標向左移動。如果游標已經在最左邊一行，則游標移動到同一列最右邊一行。
- 輸入小寫字母 `j`，游標向下移動。如果游標已經在最下面一列，則游標移動到同一行最上面一列。
- 輸入小寫字母 `k`，游標向上移動。如果游標已經在最上面一列，則游標移動到同一行最下面一列。
- 輸入小寫字母 `l`，游標向右移動。如果游標已經在最右邊一行，則游標移動到同一列最左邊一行。

作為 Vscode 資深教徒的你，擔下了實做 VSCODE 插件的重責大任。現在給你 Vim 編輯區域的高和寬，以及使用者輸入的文字，你能畫出最終編輯區域的內容嗎？

輸入格式

輸入共有兩行。第一行有兩個整數 H、W，分別代表編輯器中編輯區域的高和寬。第二行包含一個由大小寫英文字母組成的字串 s，代表使用者在編輯器中依序輸入的按鍵。

輸出格式

輸出 H 行，每行 W 個字元，代表編輯器版面上最後留下的文字。如果某一格到最後還是空白，請輸出一個點（`.`）。

資料範圍

- $1 \le H \times W \le 10^6$
- $1 \le |s| \le 5 \times 10^6$
- s 僅包含大小寫英文字母。

測試範例

輸入範例 1

```
3 5
TheQuickBrownFoxJumpsOverTheLazyDog
```

輸出範例 1

```
ereLa
zyDog
mpsOv
```

輸入範例 2

```
3 5
hlkjhkjkljhkljhlkhjlkjhklhlj
```

輸出範例 2

```
.....
.....
.....
```

輸入範例 3

```
3 5
ByTheWayIUseArchwq
```

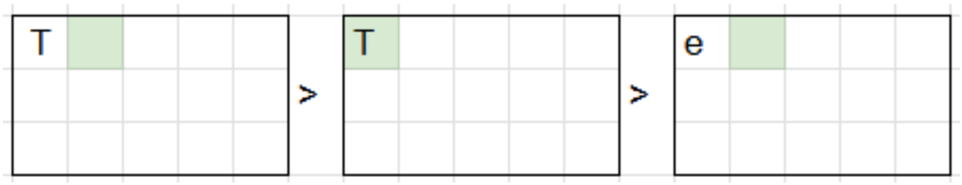
輸出範例 3

```
ByeWa
yIUse
Arwq.
```

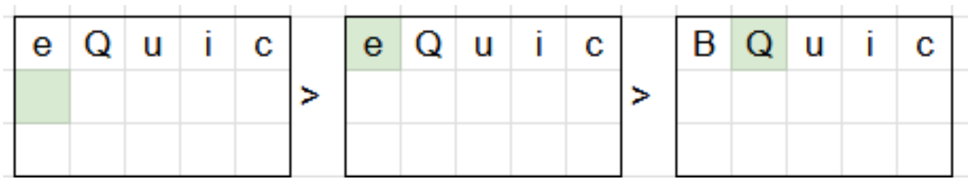
範例說明

在範例一中，輸入到各階段的編輯區內容為：

輸入到 `The` 時，這裡的 `h` 使游標向左移動一格，導致接下來輸入的 `e` 覆蓋了原本的 `T`：

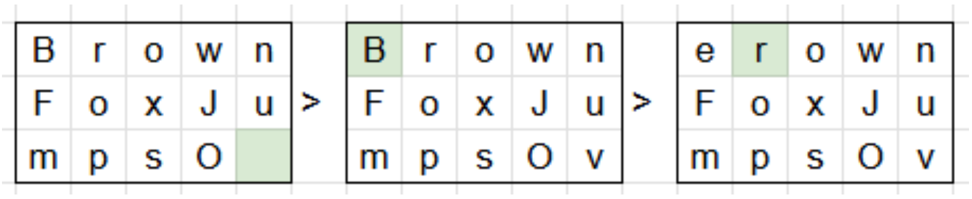


輸入到 `TheQuickB`，`k` 使游標上移，使得接下來的 `B` 回到了左上角，覆蓋了原本的 `e`：



輸入到 `TheQuickBrownFoxJu` 時，這裡的 `J` 是大寫的，因此很正常的繼續往下執行。

輸入到 `TheQuickBrownFoxJumpsOve`，輸入 `v` 後來到版面的最後一格，因此回到左上角，使得新增的 `e` 覆蓋了原本的 `B`：



接下來將剩餘的字元處理完畢後就可以得到最後的結果。

在範例二中，輸入的內容全部都是 `hjkl`，僅僅代表了移動而沒有寫入任何資料，因此最後的版面依然為空，並以 `.` 來表示空白的格子。

在範例三中，前後的兩個 `h` 分別導致 `T` 以及 `c` 被覆蓋，其餘的文字皆為正常的輸入。

> `wq` 居然沒有用，究竟要怎麼退出 Vim 呢？

2_Vim vs Code

(10 points)

Time Limit: 1 second

Memory Limit: 256 MB

Statement

Which do you prefer, Vim or VSCode?

The debate between Vim and VSCode is not only a choice between editors, but a battle of tradition and modernity. Programmers have formed two factions: the Vim faction and the VSCode faction, each trying to prove that their editor is superior to the other.

One day, the VSCode followers came up with an evil plan. They believed that it is so absurd that Vim distinguishes between normal mode and insert mode. Who on earth uses HJKL as arrow keys? Therefore, they installed a plugin, Vim Suspiciously Compromised On Direction-keys Entered (abbreviated as the VSCODE plugin), into Vim editors. Vim editors with this plugin installed misinterpret some letters as arrow keys, preventing users from typing normally and making them realize how ridiculous Vim's design is.

To be specific, Vim provides a grid as its editing area. The grid is H cells high and W cells wide, which is $H \times W$ cells in total. Each cell stores a character that may be edited according to the user's input. Additionally, there is a cursor indicating the cell being edited. At the beginning, every cell is initialized with a space character, and the cursor is placed at the top-left corner. When the user enters a key, the cell under the cursor will be overwritten with that key, and the cursor moves one cell to the right; if the cursor is already at the rightmost column, it moves to the leftmost cell in the row below; if the cursor is at the bottom-right corner, it returns to the top-left corner. However, due to the VSCODE plugin, if one of the following four characters is entered, they will be processed like arrow keys without modifying the editing area:

- lowercase `h`: cursor moves left. If the cursor is already at the leftmost column, it moves to the rightmost cell on the same row.
- lowercase `j`: cursor moves down. If the cursor is already at the bottommost row, it moves to the topmost cell on the same column.
- lowercase `k`: cursor moves up. If the cursor is already at the topmost row, it moves to the bottommost cell on the same column.
- lowercase `l`: cursor moves right. If the cursor is already at the rightmost column, it moves to the leftmost cell on the same row.

As a senior VSCode follower, you are in charge of implementing the VSCODE plugin. Given the dimensions of Vim's editing area, along with the user's input text, can you determine the final state of the editing area?

Input Format

A testcase comprises two lines. The first line contains two integers, H and W, representing the height and the width of Vim's editing area. The second line contains a single string s consisting of uppercase and lowercase English letters, which represents the Vim user's input.

Output Format

Output H lines, each containing W characters, representing the characters left on the editing area at the end. For the cells remaining a space character at the end, please print a period (`.`) to indicate them.

Constraints

- $1 \leq H \times W \leq 10^6$
- $1 \leq |s| \leq 5 \times 10^6$
- s comprises uppercase and lowercase English letters only.

Test Cases

Input 1

```
3 5
TheQuickBrownFoxJumpsOverTheLazyDog
```

Output 1

```
ereLa
zyDog
mpsOv
```

Input 2

```
3 5
hlkjhkjkljhkljhlkhjlkjhklhlj
```

Output 2

```
.....
.....
.....
```

Input 3

```
3 5
ByTheWayIUseArchwq
```

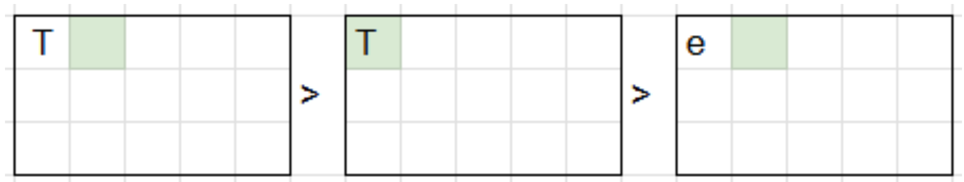
Output 3

```
ByeWa
yIUse
Arwq.
```

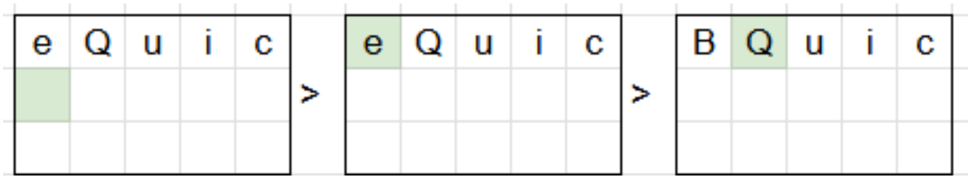
Illustrations

In the example 1,

When we process `The`, the `h` will make the cursor go left instead of putting the `h` into the editor area. The next input, e, will then overwrite the original `T`:

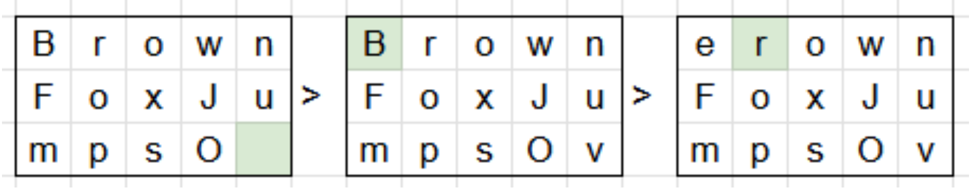


When we process `TheQuickB`, the `k` here will make the cursor go up instead of putting the `k` into the editor area. The next input `B` will then overwrite the original `e`:



When we process to `TheQuickBrownFoxJu`, the `J` is capitalized, so we see it as normal input, putting the `J` into editor area.

When we process to `TheQuickBrownFoxJumpsOve`, the input `v` is at the last slot of the editor area, so the cursor will go back to the first slot of the editor area. The `e` will then overwrite the original `B`:



We can then process the remaining input normally to get the final result.

In example 2, the input is composed of `hjkl`, which are all special characters used to move the cursor. Hence, the final editor area remains empty, and we use `.` to represent the empty slots.

In example 3, the two `h`s in the input will cause `T` and `c` to be overwritten. The remaining input consists of normal character.

> `wq` is not working! How can you exit Vim?

3_子序列構造(Subsequence Construction)

(15 分)

時間限制: 1 second
記憶體限制: 256 MB

題目敘述

現在有一個長度為 n 的序列 $a_1, a_2, \ldots, a_n$，請找出最大的正整數 m 使得存在長度為 m 的序列 $b_1, b_2, \ldots, b_m$ 滿足：

- $1, 2, \ldots, m$ 在 $b_1, b_2, \ldots, b_m$ 中各出現一次。
- $b_1, b_2, \ldots, b_m$ 是 $a_1, a_2, \ldots, a_n$ 的子序列，也就是說，可以透過從序列 $a_1, a_2, \ldots, a_n$ 中刪除若干個元素得到 $b_1, b_2, \ldots, b_m$。

或是回報滿足要求的 m 和 $b_1, b_2, \ldots, b_m$ 不存在。

如果滿足要求的 m 存在，請同時找出對應的一個序列 $b_1, b_2, \ldots, b_m$。

輸入格式

第一行輸入一個正整數 n。

第二行輸入 n 個正整數 $a_1, a_2, \ldots, a_n$。

輸出格式

如果滿足要求的 m 不存在，則輸出一行，這行輸出一個 0 即可。

如果滿足要求的 m 存在，則輸出兩行。

第一行輸出一個正整數 m。

第二行輸出 m 個正整數 $b_1, b_2, \ldots, b_m$。如果有不只一種序列 $b_1, b_2, \ldots, b_m$ 滿足條件，輸出任意一個就好。

注意，m 必須要是滿足要求的最大值。

資料範圍

- $1 \leq n \leq 2 \times 10^5$
- $1 \leq a_i \leq 2 \times 10^5$

測試範例

輸入範例 1

```
5
3 6 2 3 1
```

輸出範例 1

```
3
3 2 1
```

輸入範例 2

```
4
8 7 6 3
```

輸出範例 2

```
0
```

輸入範例 3

```
10
1 10 2 9 3 8 4 7 5 6
```

輸出範例 3

```
10
1 10 2 9 3 8 4 7 5 6
```

範例說明

範例 1 中，最大能滿足條件的 m 是 3，而序列 $[3, 2, 1]$ 和 $[2, 3, 1]$ 皆滿足題目要求，輸出其中一個就好。

序列 $[1, 3, 2]$ 不滿足條件，因為它不是 $[3, 6, 2, 3, 1]$ 的子序列。

序列 $[3, 3, 1]$ 也不滿足條件，因為 $1, 2, 3$ 並沒有在 $[3, 3, 1]$ 中各出現一次。

範例 2 中，不存在滿足條件的 m。

範例 3 中，最大的 m 是 10，也就是說取整個序列 a 會滿足條件。

3_Subsequence Construction

(15 points)

Time Limit: 1 second
Memory Limit: 256MB

Statement

Given a sequence of length n, denoted as $a_1, a_2, \ldots, a_n$, find the largest positive integer m such that there exists a sequence of length m, denoted as $b_1, b_2, \ldots, b_m$, such that:

- Each of $1, 2, \ldots, m$ appears exactly once in $b_1, b_2, \ldots, b_m$.
- The sequence $b_1, b_2, \ldots, b_m$ is a subsequence of $a_1, a_2, \ldots, a_n$, meaning it can be obtained by deleting some elements from the sequence $a_1, a_2, \ldots, a_n$.

If no such m and $b_1, b_2, \ldots, b_m$ exist, report that they do not exist.

If m does exist, please find a corresponding sequence $b_1, b_2, \ldots, b_m$ aswell.

Input Format

The first line contains a positive integer n.

The second line contains n positive integers $a_1, a_2, \ldots, a_n$.

Output Format

If no such m exists, output a single line containing the integer 0.

If such an m exists, output two lines:

The first line should contain the positive integer m.

The second line should contain m positive integers $b_1, b_2, \ldots, b_m$. If there are multiple valid sequences, output any one of them.

Note that m must be the maximum possible value satisfying the conditions.

Constraints

- $1 \leq n \leq 2 \times 10^5$
- $1 \leq a_i \leq 2 \times 10^5$

Test Cases

Input 1

```
5
3 6 2 3 1
```

Output 1

```
3
3 2 1
```

Input 2

```
4
8 7 6 3
```

Output 2

```
0
```

Input 3

```
10
1 10 2 9 3 8 4 7 5 6
```

Output 3

```
10
1 10 2 9 3 8 4 7 5 6
```

Illustrations

In Example 1, the maximum value of m that satisfies the conditions is 3, and the sequence $[3, 2, 1]$ meets the requirements. The sequence $[2, 3, 1]$ also meets the requirements, and either can be output.

The sequence $[1, 3, 2]$ doesn't meet the requirements because it is not a subsequence of $[3, 6, 2, 3, 1]$.

The sequence $[3, 3, 1]$ also doesn't meet the requirements because some of $1, 2, 3$ doesn't not appear exactly once in $[3, 3, 1]$.

In Example 2, no such m exists that meets the conditions.

In Example 3, the largest value of m is 10, meaning the entire sequence a meets the conditions.

4_字串遊戲 (String Game)

(15 分)

時間限制: 1 second

記憶體限制: 256 MB

題目敘述

呱呱一號與呱呱二號想要來一場經典的字串遊戲，這個遊戲由兩個玩家輪流進行，每一次遊戲都會有一個字串 s，每一個玩家的回合開始時都必須選擇三個正整數 $(i, j, k), 1 \leq i < j < k \leq |s|$ 滿足 $s_i =$"a", $s_j = s_k =$"b"。

如果之前的某一回合已經選過這三個正整數了，那麼接下來的回合就不能再選擇這三個正整數，舉個例子來說如果某一回合選擇了 $(1, 2, 4)$，那麼接下來就不能再選擇 $(1, 2, 4)$，但可以選擇 $(1, 2, 5)$。

遊戲的結束條件為，如果某一回合該玩家無法選出三個滿足條件的正整數，則該玩家輸了這一場遊戲。

已知呱呱一號跟呱呱二號都會使用最佳策略來玩這場遊戲，請你設計一個程式，給定一個字串 s，並輸出如果呱呱一號是先手玩家，那麼誰會贏下這一場遊戲。

輸入格式

輸入一個字串 s，意義如同題目敘述所述。

輸出格式

如果呱呱一號會贏下這一場遊戲則輸出 "DuckDuck 1"，反之則輸出 "DuckDuck 2"（不含引號）。

資料範圍

- $|s| \leq 100$。
- s 中只會有小寫字母。

測試範例

輸入範例 1

```
ytp
```

輸出範例 1

```
DuckDuck 2
```

輸入範例 2

```
abb
```

輸出範例 2

```
DuckDuck 1
```

輸入範例 3

```
cabbabbc
```

輸出範例 3

```
DuckDuck 1
```

範例說明

在範例三中，可以選擇的 i, j, k 有：

- $(2, 3, 4)$
- $(2, 3, 7)$
- $(2, 3, 8)$
- $(2, 4, 7)$
- $(2, 4, 8)$
- $(2, 7, 8)$
- $(5, 7, 8)$

共 7 組，因此呱呱一號獲得勝利。

4_String Game

(15 points)

Time Limit: 1 second

Memory Limit: 256 MB

Statement

DuckDuck No.1 and DuckDuck No.2 want to play a classic string game. The game is turn-based, and each game starts with a string s. At the beginning of each player's turn, they must choose three positive integers (i, j, k), where $1 \leq i < j < k \leq |s|$, and these integers must satisfy $s_i =$"a", $s_j =$"b", $s_k =$"b".

If a set of these three integers has already been chosen in a previous turn, it cannot be selected again. For example, if $(1, 2, 4)$ was chosen in one turn, it cannot be chosen again, but $(1, 2, 5)$ could be.

The game ends when a player cannot select three integers that meet the criteria, and that player loses the game.

Given that both DuckDuck No.1 and DuckDuck No.2 will play optimally, design a program that, given a string s, outputs who would win the game if DuckDuck No.1 starts first.

Input Format

The first line contains a string s, which is as described above.

Output Format

Output "DuckDuck 1" if DuckDuck No.1 would win the game, otherwise output "DuckDuck 2" (without quotes).

Constraints

- $|s| <= 100$.
- s only contains lowercase alphabets.

Test Cases

Input1

```
ytp
```

Output 1

```
DuckDuck 2
```

Input 2

```
abb
```

Output 2

```
DuckDuck 1
```

Input 3

```
cabbabbc
```

Output 3

```
DuckDuck 1
```

Illustrations

In example 3, we can choose following (i, j, k):

- $(2, 3, 4)$
- $(2, 3, 7)$
- $(2, 3, 8)$
- $(2, 4, 7)$
- $(2, 4, 8)$
- $(2, 7, 8)$
- $(5, 7, 8)$

There are 7 choices in total, hence DuckDuck No.1 will win the game.

5_球球遊戲(Ball Game)

(7 分 / 13 分)

時間限制: 1 second
記憶體限制: 256 MB

題目敘述

小櫻與小桃正在玩一個遊戲。一開始桌子上一共有 N 顆球，由左到右上面依序寫著 $1, 2, \ldots, N$。小桃會依序進行 M 次操作，操作會是以下兩種之一：

- 小桃會將從左邊數過來第 a 顆球與第 b 顆球交換
- 小桃會給小櫻與第 c 顆球上寫的數字相同數量的錢

現在小桃告訴你她依序會進行哪些操作，請告訴她小櫻總共會拿到多少錢。

輸入格式

輸入第一行有兩個正整數 N, M，代表有幾顆球與有幾次操作。

接下來 M 行每行會代表一種操作，每一行會是以下兩種格式之一：

- $1\ a_i\ b_i$，代表會將第 a_i 顆球與第 b_i 顆球交換
- $2\ c_i$，代表會給小櫻等於第 c_i 顆球寫的數字數量的錢

注意小桃會依序進行這 M 個操作。

輸出格式

請輸出一行，該行有一個整數，代表小櫻總共會拿到多少錢。

資料範圍

- $2 \leq N \leq 10^9$
- $1 \leq M \leq 10^5$
- $1 \leq a_i, b_i, c_i \leq N$
- $a_i \neq b_i$

子任務

- 子任務 1 滿足 $2 \leq N \leq 10^5$ (7 分)
- 子任務 2 沒有額外限制 (13 分)

測試範例

輸入範例 1

```
4 5
1 2 3
2 2
1 3 4
1 1 4
2 1
```

輸出範例 1

```
5
```

輸入範例 2

```
864197532 2
2 48763
2 56562
```

輸出範例 2

```
105325
```

範例說明

在第一筆範例測資中，一開始球的編號依序為 $[1, 2, 3, 4]$，之後：

- 第一個操作後，球的編號會變成 $[1, 3, 2, 4]$
- 第二個操作小桃會給小櫻 3 塊錢
- 第三個操作後，球的編號會變成 $[1, 3, 4, 2]$
- 第四個操作後，球的編號會變成 $[2, 3, 4, 1]$
- 第五個操作小桃會給小櫻 2 塊錢

因此答案為 5。

5_Ball Game

(7 points / 13 points)

Time Limit: 1 second
Memory Limit: 256 MB

Statement

Sakura and Momo are playing a game. Initially, there are N balls on the table, numbered sequentially $1, 2, \ldots, N$ from left to right. Momo will perform M operations in sequence, which can be one of the following two types:

- Swap the ball at position a with the ball at position b
- Give Sakura an amount of money equal to the number written on the c-th ball

Now, Momo tells you the sequence of operations she will perform. Please tell her how much money Sakura will earn in total.

Input Format

The first line contains two positive integers N and M, representing the number of balls and the number of operations.

The next M lines each represent an operation, which can be in one of the following two formats:

- $1\ a_i\ b_i$, which means swapping the ball at position a_i with the ball at position b_i
- $2\ c_i$, which means giving Sakura the amount of money equal to the number on the ball at position c_i

Note that Momo will perform these M operations in sequence.

Output Format

Output a single integer, representing the total amount of money Sakura will earn.

Constraints

- $2 \leq N \leq 10^9$
- $1 \leq M \leq 10^5$
- $1 \leq a_i, b_i, c_i \leq N$
- $a_i \neq b_i$

Subtasks

- Subtask 1: $2 \leq N \leq 10^5$ (7 points)
- Subtask 2: No additional constraints (13 points)

Test Cases

Input 1

```
4 5
1 2 3
2 2
1 3 4
1 1 4
2 1
```

Output 1

```
5
```

Input 2

```
864197532 2
2 48763
2 56562
```

Output 2

```
105325
```

Illustrations

In the first example, initially the balls are numbered $[1, 2, 3, 4]$. After:

- The first operation, the balls are $[1, 3, 2, 4]$.
- The second operation, Momo gives Sakura 3 dollars.
- The third operation, the balls are $[1, 3, 4, 2]$.
- The fourth operation, the balls are $[2, 3, 4, 1]$.
- The fifth operation, Momo gives Sakura 2 dollars.

Thus, the total money Sakura earns is 5 dollars.

6_配對遊戲 (Matching Game)

(20 分)

時間限制: 1 second
記憶體限制: 256 MB

題目敘述

座標平面上有 $2n$ 個點，其中有 n 個紅點和 n 個藍點，第 i 個紅點的座標是 $(0, a_i)$，第 i 個藍點的座標則是 (L, b_i)。所有的紅點都在 $x = 0$ 的垂直線上，所有的藍點則都在 $x = L$ 的垂直線上。注意，點的座標有可能會重複，但他們不算同一個點。

一開始，所有點都沒有被配對過，而你的起始分數為 0。

在一次配對操作中，你可以選擇尚未被配對過的紅點和藍點各一個並將它們配對，假設兩個點之間的歐幾里得距離為 d，配對後你獲得的分數就是 d。兩個點 $(x_1, y_1), (x_2, y_2)$ 之間的歐幾里得距離為 $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$。

你的目標是利用 n 次配對操作獲得最多的分數。請寫一支程式，算出你能獲得的最大分數。

輸入格式

第一行輸入兩個正整數 n, L。

第二行輸入 n 個整數 $a_1, a_2, \ldots, a_n$，代表 n 個紅點的座標為 $(0, a_1), (0, a_2), \ldots, (0, a_n)$。

第三行輸入 n 個整數 $b_1, b_2, \ldots, b_n$，代表 n 個藍點的座標為 $(L, b_1), (L, b_2), \ldots, (L, b_n)$。

輸出格式

輸出一個實數代表你能獲得的最大分數。

你的回答只要絕對誤差或相對誤差不超過 10^{-6} 就會視為正確，也就是說假設你的回答為 A 而正確答案為 B，則你的答案會視為正確若 $\frac{|A-B|}{\max(1,|B|)} \leq 10^{-6}$。

資料範圍

- $1 \leq n \leq 10^5$
- $1 \leq L \leq 10^9$
- $-10^9 \leq a_i, b_i \leq 10^9$

測試範例

輸入範例 1

```
3 1
4 5 1
2 6 3
```

輸出範例 1

```
9.675510736134259
```

輸入範例 2

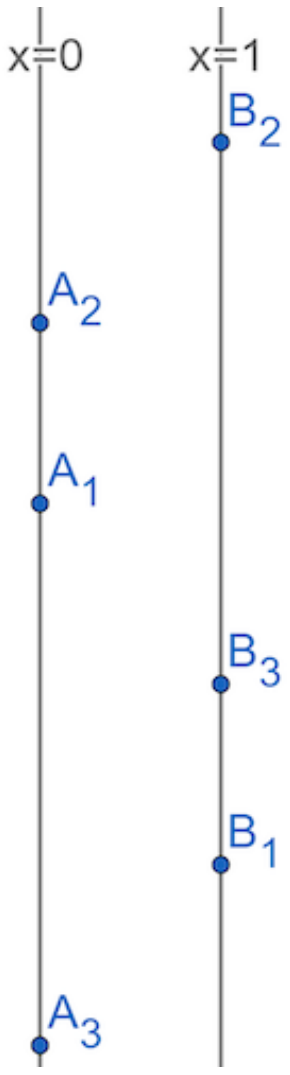
```
5 2
1 1 1 1 1
1 1 1 1 1
```

輸出範例 2

```
10.000
```

範例說明

範例一中，令 A_i 為點 $(0, a_i)$，B_i 為點 $(1, b_i)$，則最好的配對方法是將紅點 A_1, A_2, A_3 依序和藍點 B_3, B_1, B_2 配對。下圖為這六個點和兩條垂直線的相對位置。



範例二中，不管怎麼配對，最後的分數一定是 10。

6_Matching Game

(20 points)

Time Limit: 1 second
Memory Limit: 256MB

Statement

There are $2n$ points on a coordinate plane, consisting of n red points and n blue points. The coordinates of the i-th red point are $(0, a_i)$, and the coordinates of the i-th blue point are (L, b_i). All red points lie on the vertical line $x = 0$, and all blue points lie on the vertical line $x = L$. Note that the coordinates of the points may be repeated, but they are still considered distinct points.

Initially, none of the points are paired, and your starting score is 0.

In one pairing operation, you can choose one unpaired red point and one unpaired blue point and pair them together. If the Euclidean distance between the two points is d, then you earn a score of d after pairing them. The Euclidean distance between two points $(x_1, y_1), (x_2, y_2)$ is $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

Your goal is to achieve the highest possible score after performing n pairing operations. Write a program to calculate the maximum score you can obtain.

Input Format

The first line contains two positive integers n and L.

The second line contains n integers $a_1, a_2, \ldots, a_n$, representing the coordinates of the n red points $(0, a_1), (0, a_2), \ldots, (0, a_n)$.

The third line contains n integers $b_1, b_2, \ldots, b_n$, representing the coordinates of the n blue points $(L, b_1), (L, b_2), \ldots, (L, b_n)$.

Output Format

Output a real number representing the maximum score you can achieve.

Your answer will be considered correct if the absolute or relative error does not exceed 10^{-6}, meaning that if your answer is A and the correct answer is B, your answer will be considered correct if $\frac{|A-B|}{\max(1,|B|)} \leq 10^{-6}$.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq L \leq 10^9$
- $-10^9 \leq a_i, b_i \leq 10^9$

Test Cases

Input 1

```
3 1
4 5 1
2 6 3
```

Output 1

```
9.675510736134259
```

Input 2

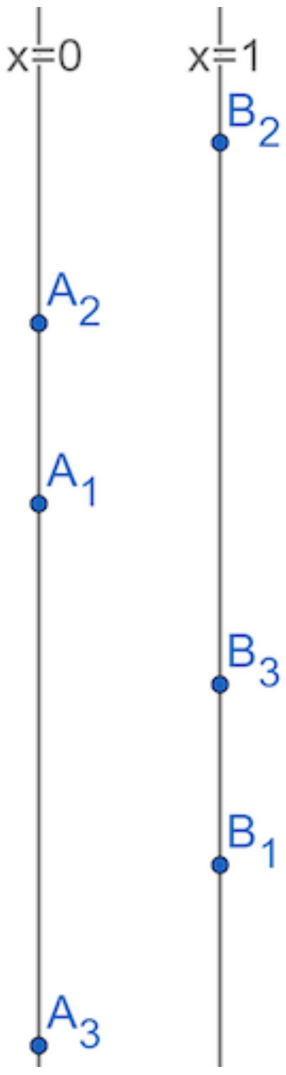
```
5 2
1 1 1 1 1
1 1 1 1 1
```

Output 2

```
10.000
```

Illustrations

In the first example, let A_i be the point $(0, a_i)$ and B_i be the point $(1, b_i)$.
The best pairing method is to pair the red points A_1, A_2, A_3 with the blue points B_3, B_1, B_2, respectively.
The following picture shows the relative positions of the six points and two vertical lines.



In the second example, no matter how you pair the points, the final score will always be 10.

7_無趣的資料結構(Boring Data Structure)

(3 分/ 6 分/ 11 分)

時間限制: 1 second
記憶體限制: 256 MB

題目敘述

樂奈最近迷上了函數，特別是一個名為 mex（**m**inimum **ex**cluded value）的函數。mex 函數的定義如下：

- 對於一個集合 S，$\text{mex}(S)$ 是最小的非負整數 x 滿足 S 中不存在 x。
    - 舉例來說：$\text{mex}(\{0,1,3,5\}) = 2$、$\text{mex}(\{1,3\}) = 0$、$\text{mex}(\{0,1,2\}) = 3$。

樂奈在練新歌之餘，用這個函數構造了一道資料結構問題。給定一個數字 n 和 n 個集合 $A_1, A_2, \ldots, A_n$，其中第 i 個集合包含了 $0, 1, 2, \ldots, n-1$ 之中**除了** $e_{i,1}, e_{i,2}, \ldots, e_{i,k_i}$ **之外的所有整數**。樂奈想設計一個能支援 q 次以下兩種操作的資料結構：

1. `mex-add`：令 $A =_{\text{def}} A_1 \cap A_2 \cap \cdots \cap A_n$ 是這 n 個集合的交集且 $x =_{\text{def}} \text{mex}(A)$。對 $i = 1, 2, \ldots, n$，如果 x 不在 A_i 之中，則將 x 加入 A_i。
2. `toggle` x：對 $i = 1, 2, \ldots, n$，如果 x 在 A_i 之中，則將其從 A_i 移除；反之，則將 x 加入 A_i。

請在每次操作完之後輸出所有集合內的數字之和。

樂奈在構造完這個問題之後，認為這個問題實在太無趣就跑去吃抹茶巴菲了。喵喵在練吉他的時候無意間看到了留在練習室的題目，於是就順手把它丟給了你。你能解開這個問題嗎？~~還是你要再把問題丟給你的隊友來做呢？~~

P.S. 交集的定義：x 存在於 $S_1 \cap S_2 \cap \cdots \cap S_k$ 之中若且唯若 x 同時存在於所有 $S_1, S_2, \ldots, S_k$ 之中。

輸入格式

- line 1: $n \quad q$
- line $1+i$ $(1 \le i \le n)$: $k_i \quad e_{i,1} \quad e_{i,2} \quad \cdots \quad e_{i,k_i}$
- line $n+1+j$ $(1 \le j \le q)$: Query_j

其中，Query_j 的格式為下列兩者之一：

- `mex-add`
- `toggle` x_j

以上的變數所代表的含意都如同題目敘述所述。

輸出格式

- line 1: ans

資料範圍

- $1 \le n \le 300\,000$。
- $1 \le q \le 300\,000$。
- $0 \le k_i \le n$ （$1 \le i \le n$）。
    - $\sum_{i=1}^{n} k_i \le 1\,000\,000$。
- $0 \le e_{i,1} < e_{i,2} < \cdots < e_{i,k_i} \le n-1$ （$1 \le i \le n$）。
- $op_j \in \{\texttt{mex-add}, \texttt{toggle}\}$ （$1 \le j \le q$）。
- $0 \le x_j \le n-1$ （$1 \le j \le q$ 且 $op_j = \texttt{toggle}$）。
- 輸入的數字都是整數。

子任務

1. (3 points) $op_j = \texttt{toggle}$ （$1 \le j \le q$）。
2. (6 points) $k_i = n$ （$1 \le i \le n$）。
3. (11 points) 無額外限制。

測試範例

輸入範例 1

```
5 8
1 3
1 2
4 0 1 2 3
1 2
3 0 1 4
toggle 4
mex-add
mex-add
toggle 4
toggle 2
mex-add
toggle 3
mex-add
```

該範例輸入符合子任務 3 的限制。

輸出範例 1

```
20
20
22
34
36
40
37
46
```

- 最一開始：
  - $A_1 = \{0, 1, 2, 4\}$
  - $A_2 = \{0, 1, 3, 4\}$
  - $A_3 = \{4\}$
  - $A_4 = \{0, 1, 3, 4\}$
  - $A_5 = \{2, 3\}$
- 第一次操作 $\text{Query}_1 = \texttt{toggle}\ \ 4$ 之後：
  - $A_1 = \{0, 1, 2\}$（移除 4）
  - $A_2 = \{0, 1, 3\}$（移除 4）
  - $A_3 = \{\}$（移除 4）
  - $A_4 = \{0, 1, 3\}$（移除 4）
  - $A_5 = \{2, 3, 4\}$（加入 4）
  - 數字之和為 $(0+1+2)+(0+1+3)+(0+1+3)+(2+3+4)=20$。
- 第二次操作 $\text{Query}_2 = \texttt{mex-add}$ 之後：
  - $A_1 = \{0, 1, 2\}$
  - $A_2 = \{0, 1, 3\}$
  - $A_3 = \{0\}$（加入 0）
  - $A_4 = \{0, 1, 3\}$
  - $A_5 = \{0, 2, 3, 4\}$（加入 0）
  - 數字之和為 $(0+1+2)+(0+1+3)+(0)+(0+1+3)+(0+2+3+4)=20$。
- 第三次操作 $\text{Query}_3 = \texttt{mex-add}$ 之後：
  - $A_1 = \{0, 1, 2\}$
  - $A_2 = \{0, 1, 3\}$
  - $A_3 = \{0, 1\}$
  - $A_4 = \{0, 1, 3\}$
  - $A_5 = \{0, 1, 2, 3, 4\}$
  - 數字之和為 $(0+1+2)+(0+1+3)+(0+1)+(0+1+3)+(0+1+2+3+4)=22$。

輸入範例 2

```
3 8
2 0 2
1 2
0
toggle 0
toggle 1
toggle 0
toggle 2
toggle 0
toggle 1
toggle 0
toggle 2
```

該範例輸入符合子任務 1, 3 的限制。

輸出範例 2

```
5
2
2
4
4
7
7
5
```

輸入範例 3

```
4 10
4 0 1 2 3
4 0 1 2 3
4 0 1 2 3
4 0 1 2 3
toggle 2
mex-add
toggle 1
mex-add
toggle 0
toggle 3
mex-add
toggle 2
mex-add
mex-add
```

該範例輸入符合子任務 2, 3 的限制。

輸出範例 3

```
8
8
12
24
24
12
12
4
12
24
```

7_Boring Data Structure

(3 points/6 points/11 points)

Time Limit: 1 second
Memory Limit: 256MB

Statement

Raana-chan has recently become obsessed with functions, especially a function called mex (**m**inimum **ex**cluded value). The mex function is defined as follows:

- For a set S, $\text{mex}(S)$ is the smallest non-negative integer x such that x is not in S.
  - For example: $\text{mex}(\{0, 1, 3, 5\}) = 2$, $\text{mex}(\{1, 3\}) = 0$, $\text{mex}(\{0, 1, 2\}) = 3$.

Raana-chan has constructed a data structure problem using this function in her spare time while practicing a new song. Given a number n and n sets $A_1, A_2, \ldots, A_n$, where the i^{th} set contains all integers from 0 to $n-1$ **except** $e_{i,1}, e_{i,2}, \ldots, e_{i,k_i}$. Raana-chan want to design a data structure which supports the following two operations q times:

1. `mex-add`: Let $A =_{\text{def}} A_1 \cap A_2 \cap \cdots \cap A_n$ be the intersection of these n sets and $x =_{\text{def}} \text{mex}(A)$. For $i = 1, 2, \ldots, n$, if x is not in A_i, then add x to A_i.
2. `toggle` x: For $i = 1, 2, \ldots, n$, if x is in A_i, then remove it from A_i; otherwise, add x to A_i.

Please output the sum of all numbers in the sets after each operation.

Raana-chan thought this problem was too boring and she went to eat matcha parfait. As the next user of the practice room, Neko-chan accidentally saw the problem left in the room while practicing guitar, and handed it to you. Can you solve this problem? ~~Or do you want to hand the problem to your teammates again?~~

P.S. Definition of intersection: x exists in $S_1 \cap S_2 \cap \cdots \cap S_k$ if and only if x exists in all $S_1, S_2, \ldots, S_k$.

Input Format

- line 1: $\quad n \;\; q$
- line $1+i$ $(1 \le i \le n)$: $\quad k_i \;\; e_{i,1} \;\; e_{i,2} \;\; \cdots \;\; e_{i,k_i}$
- line $n+1+j$ $(1 \le j \le q)$: $\quad \text{Query}_j$

Where Query_j is in the following format:

- `mex-add`
- `toggle` x_j

All variable is as described above.

Output Format

- line 1: $\quad ans$

Constraints

- $1 \le n \le 300\,000$.
- $1 \le q \le 300\,000$.
- $0 \le k_i \le n$ ($1 \le i \le n$).
  - $\sum_{i=1}^{n} k_i \le 1\,000\,000$.
- $0 \le e_{i,1} < e_{i,2} < \cdots < e_{i,k_i} \le n-1$ ($1 \le i \le n$).
- $op_j \in \{\texttt{mex-add}, \texttt{toggle}\}$ ($1 \le j \le q$).
- $0 \le x_j \le n-1$ ($1 \le j \le q$ and $op_j = \texttt{toggle}$).
- All numbers in the input are integers.

Subtasks

1. (3 points) $op_j = \texttt{toggle}$ ($1 \le j \le q$).
2. (6 points) $k_i = n$ ($1 \le i \le n$).
3. (11 points) No additional constraints.

Test Cases

Input 1

```
5 8
1 3
1 2
4 0 1 2 3
1 2
3 0 1 4
toggle 4
mex-add
mex-add
toggle 4
toggle 2
mex-add
toggle 3
mex-add
```

This sample input satisfies the constraints of Subtask 3.

Output 1

```
20
20
22
34
36
40
37
46
```

- In the beginning:
  - $A_1 = \{0, 1, 2, 4\}$
  - $A_2 = \{0, 1, 3, 4\}$
  - $A_3 = \{4\}$
  - $A_4 = \{0, 1, 3, 4\}$
  - $A_5 = \{2, 3\}$
- After $\text{Query}_1 = \texttt{toggle}\ \ 4$:
  - $A_1 = \{0, 1, 2\}$ (remove 4)
  - $A_2 = \{0, 1, 3\}$ (remove 4)
  - $A_3 = \{\}$ (remove 4)
  - $A_4 = \{0, 1, 3\}$ (remove 4)
  - $A_5 = \{2, 3, 4\}$ (add 4)
  - The sum is $(0 + 1 + 2) + (0 + 1 + 3) + (0 + 1 + 3) + (2 + 3 + 4) = 20$.
- After $\text{Query}_2 = \texttt{mex-add}$:
  - $A_1 = \{0, 1, 2\}$
  - $A_2 = \{0, 1, 3\}$
  - $A_3 = \{0\}$ (add 0)
  - $A_4 = \{0, 1, 3\}$
  - $A_5 = \{0, 2, 3, 4\}$ (add 0)
  - The sum is $(0 + 1 + 2) + (0 + 1 + 3) + (0) + (0 + 1 + 3) + (0 + 2 + 3 + 4) = 20$.
- After $\text{Query}_3 = \texttt{mex-add}$:
  - $A_1 = \{0, 1, 2\}$
  - $A_2 = \{0, 1, 3\}$
  - $A_3 = \{0, 1\}$
  - $A_4 = \{0, 1, 3\}$
  - $A_5 = \{0, 1, 2, 3, 4\}$
  - The sum is $(0 + 1 + 2) + (0 + 1 + 3) + (0 + 1) + (0 + 1 + 3) + (0 + 1 + 2 + 3 + 4) = 22$.

Input 2

```
3 8
2 0 2
1 2
0
toggle 0
toggle 1
toggle 0
toggle 2
toggle 0
toggle 1
toggle 0
toggle 2
```

This sample input satisfies the constraints of Subtasks 1, 3.

Output 2

```
5
2
2
4
4
7
7
5
```

Input 3

```
4 10
4 0 1 2 3
4 0 1 2 3
4 0 1 2 3
4 0 1 2 3
toggle 2
mex-add
toggle 1
mex-add
toggle 0
toggle 3
mex-add
toggle 2
mex-add
mex-add
```

This sample input satisfies the constraints of Subtasks 2, 3.

Output 3

```
8
8
12
24
24
12
12
4
12
24
```