少年圖靈計畫
Young Turing Program

0_送分題 – Hello World

(30分)

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!
```

0_Hello World

(30 points)

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_虹夏？多力多滋？(Nijika or Doritos?)

(5分)

時間限制: 1 second
記憶體限制: 256 MB

題目敘述

「要怎麼區分虹夏頭上的呆毛和多力多滋？」

為了解決這個世紀難題，波奇決定好好分析這兩者。經過縝密的觀察，她發現虹夏的呆毛可以用頂點在 $(nx_1, ny_1), (nx_2, ny_2), (nx_3, ny_3)$ 的三角形來表示，而多力多滋可以用頂點在 $(dx_1, dy_1), (dx_2, dy_2), (dx_3, dy_3)$ 的三角形來表示，另外她也發現這兩種三角形是不全等的。同時，為了更好分辨兩者的差異，她定義一個三角形為「虹夏類」若她與虹夏呆毛的三角形全等，為「多力多滋類」若她與多力多滋的三角形全等。

在聽說波奇解決了這個世紀難題後，有許多人依然分不清楚呆毛和多力多滋，他們將物品表示成了三角形，想要請波奇幫忙區分。然而這也讓波奇的社恐症又双叒叕發作了，所以她請你幫忙解決這些人的問題。對於每個人的提問，若該三角形為「虹夏類」則請輸出 `Nijika`，若為「多力多滋類」，請輸出 `Doritos`，若都不是則輸出 `None`，注意**需要按照題目敘述的大小寫輸出**。

兩個三角形**全等**若可以經由平移、旋轉與翻轉使兩個三角形的頂點座標相同。

備註：多力多滋沒有贊助 YTP 少年圖靈計劃

輸入格式

輸入第一行有六個整數 $nx_1, ny_1, nx_2, ny_2, nx_3, ny_3$，意義與題目敘述相同。

輸入第二行有六個整數 $dx_1, dy_1, dx_2, dy_2, dx_3, dy_3$，意義與題目敘述相同。

輸入第三行有一個正整數 q，代表向波奇詢問的人數。

接下來的 q 行，每一行有六個整數 $qx_{i,1}, qy_{i,1}, qx_{i,2}, qy_{i,2}, qx_{i,3}, qy_{i,3}$，代表第 i 個人詢問的三角形的三個頂點為 $(qx_{i,1}, qy_{i,1}), (qx_{i,2}, qy_{i,2}), (qx_{i,3}, qy_{i,3})$。

輸出格式

請輸出 q 行，其中第 i 行包含一個字串，代表第 i 個詢問的答案。

資料範圍

- $1 \leq q \leq 1000$
- 點座標範圍在 0 到 10^9 之間
- 三角形的三點不共線
- 虹夏呆毛三角形與多力多滋三角形不全等

測試範例

輸入範例 1

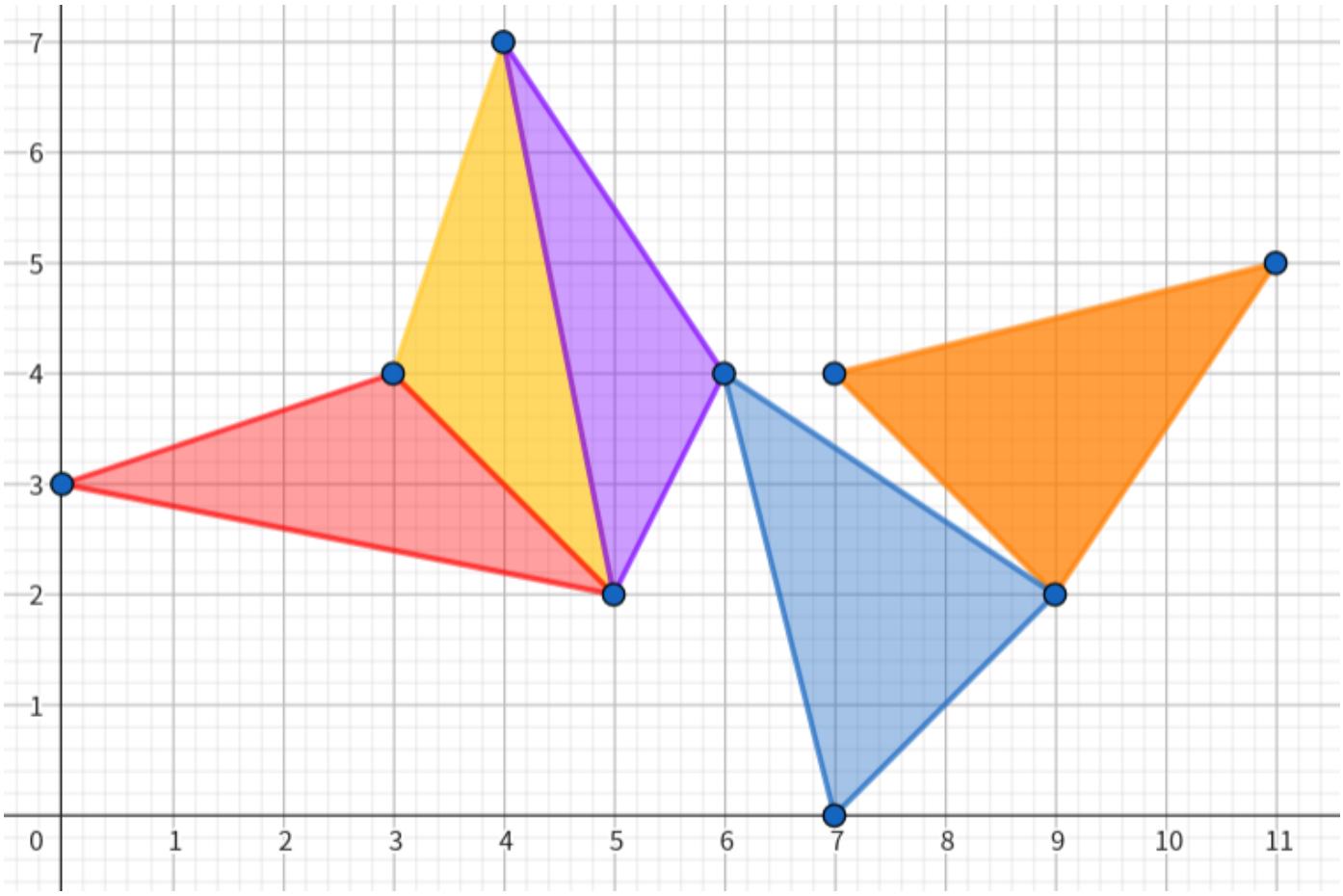
```
5 2 3 4 4 7
7 4 9 2 11 5
3
0 3 5 2 3 4
5 2 6 4 4 7
9 2 7 0 6 4
```

輸出範例 1

```
Nijika
None
Doritos
```

範例說明

以下為範例測資的圖片：



其中黃色為虹夏呆毛三角形，橘色為多力多滋三角形，紅色為第一筆詢問，紫色為第二筆詢問，藍色為第三筆詢問。

1_Nijika or Doritos?

(5 points)

Time Limit: 1 second
Memory Limit: 256MB

Statement

"How to distinguish Nijika's ahoge and Doritos?"

To solve this century-long puzzle, Bocchi decided to analyze them thoroughly. After meticulous observation, she found that Nijika's ahoge can be represented by a triangle with vertices at $(nx_1, ny_1), (nx_2, ny_2), (nx_3, ny_3)$, while Doritos can be represented by a triangle with vertices at $(dx_1, dy_1), (dx_2, dy_2), (dx_3, dy_3)$. Furthermore, she discovered that these two triangles are not congruent. Additionally, to better distinguish between the two, she defined a triangle as "Nijika-type" if it is congruent to Nijika's ahoge triangle, and "Doritos-type" if it is congruent to the Doritos triangle.

Upon hearing that Bocchi had solved this century-long puzzle, many people still struggled to distinguish between the ahoge and Doritos. They represented the items as triangles and sought Bocchi's help in distinguishing them. However, this triggered Bocchi's social anxiety once again. Therefore, she asked for your assistance in solving these people's problems. For each person's query, if the triangle is "Nijika-type", please output `Nijika`, if it is "Doritos-type", please output `Doritos`, and if neither, please output `None`. Note that the output should match the case **specified in the problem description**.

Two triangles are considered **congruent** if they can be transformed into one another by translation, rotation, and reflection.

Note: Doritos is not sponsoring Youth Turing Program.

Input Format

The first line contains six integers $nx_1, ny_1, nx_2, ny_2, nx_3, ny_3$, representing the vertices of Nijika's ahoge triangle.

The second line contains six integers $dx_1, dy_1, dx_2, dy_2, dx_3, dy_3$, representing the vertices of Doritos triangle.

The third line contains a positive integer q, indicating the number of people querying Bocchi.

The next q lines each contain six integers $qx_{i,1}, qy_{i,1}, qx_{i,2}, qy_{i,2}, qx_{i,3}, qy_{i,3}$, representing the vertices of the triangle queried by the i-th person, which are $(qx_{i,1}, qy_{i,1}), (qx_{i,2}, qy_{i,2}), (qx_{i,3}, qy_{i,3})$.

Output Format

The output should consist of q lines. Each line should contain a string representing the answer to the i-th query.

Constraints

- $1 \leq q \leq 1000$

- The coordinates of the points are between 0 and 10^9
- The three points of a triangle are not collinear
- The triangle representing Nijika's ahoge is not congruent to the triangle representing Doritos

Test Cases

Input 1

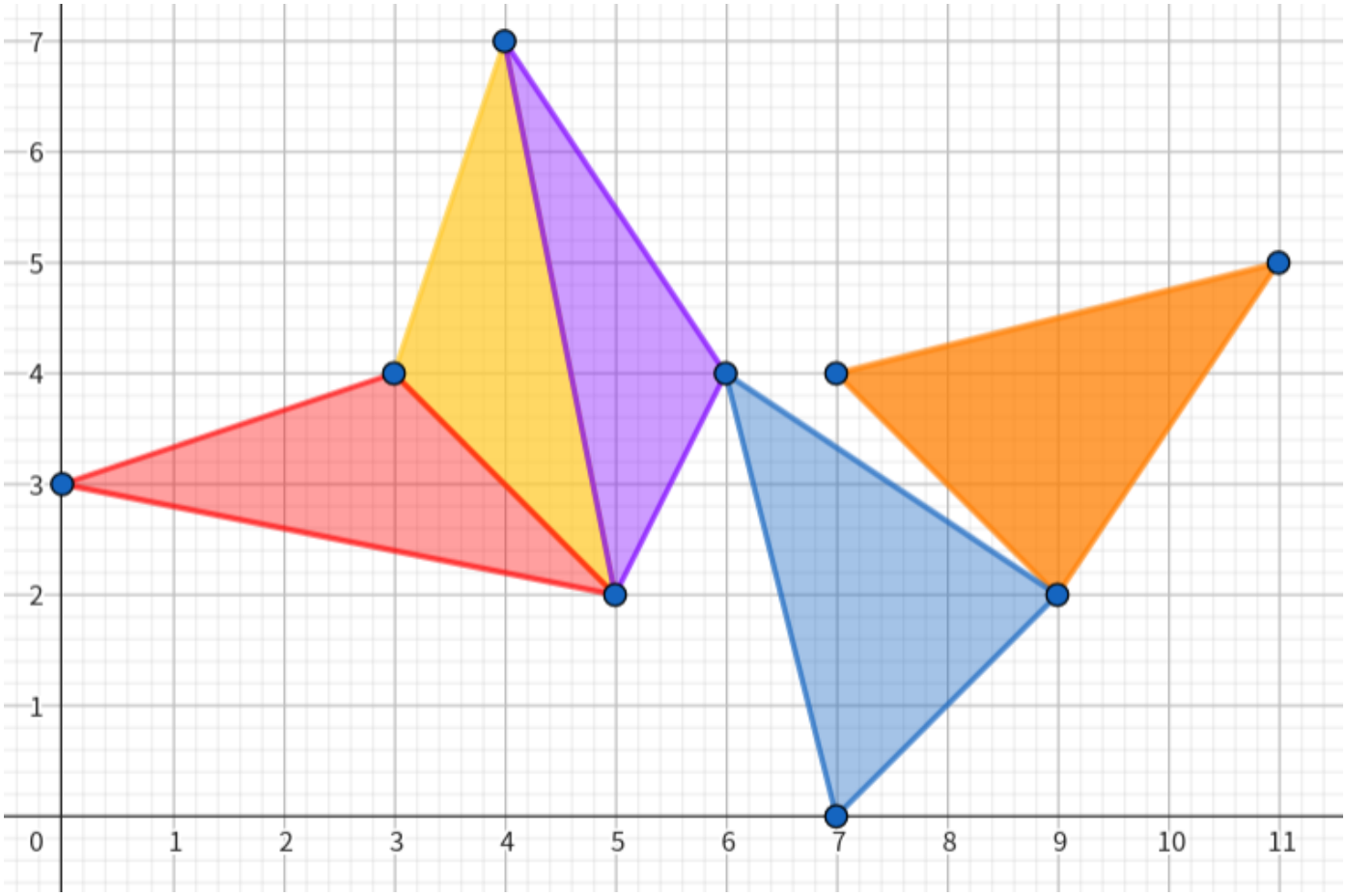
```
5 2 3 4 4 7
7 4 9 2 11 5
3
0 3 5 2 3 4
5 2 6 4 4 7
9 2 7 0 6 4
```

Output 1

```
Nijika
None
Doritos
```

Illustrations

The following is an image of the sample test case:



In the image, yellow represents the Nijika ahoge triangle, orange represents the Doritos triangle, red represents the triangle of first query, purple represents the triangle of second query, and blue represents the triangle of third query.

2_戰力指數 (War Power Analysis)

(2分 /8分)

時間限制: 1 second
記憶體限制: 256 MB

題目敘述

「富國強兵」一詞源於中國古代的典故，出自《左傳》中的一段記載。這段典故發生在戰國時期，當時晉國的國君晉文公想要實行內政改革，以強化國家的實力。他向參謀諸侯問策問道：「寡人欲富國，將何以使之？」其中一位參謀對他說：「富國之道，先強兵也。」這句話即為「富國強兵」的來源。

這句話的含義是，一個國家要想繁榮富強，首先必須建立強大的軍事實力。只有具備足夠的軍事力量，國家才能保護自身的疆土、維護國家的利益，並在外交與戰爭中取得優勢。

現在的你身兼大任，要經營一個有 n 個角色的兵隊。第 i 個角色的戰力為 $a_i \cdot x + b_i$，其中 x 為等級數且預設為 1 級，每升一級需要 c_i 的金幣。現在你擁有 w 元，試問這個兵隊的最大戰力值為何？

輸入格式

第一行有兩個正整數 n、w，代表這個兵隊有 n 個角色，以及你所擁有的金幣數量。

接下來的 n 行，每行有三個正整數 a_i、b_i、c_i，如題意所述。

輸出格式

輸出一個正整數，代表兵隊最大的戰力值。

資料範圍

- $1 \leq n \leq 1000$。
- $1 \leq w \leq 10^5$。
- $1 \leq a_i, b_i, c_i \leq 1000$（$1 \leq i \leq n$）。

子任務

- 子任務 1, $n \leq 10$。
- 子任務 2, 無額外限制。

測試範例

輸入範例 1

```
3 10
1 2 3
3 4 1
2 3 1
```

輸出範例 1

```
45
```

範例說明

在範例 1 中，把所有的金幣都給第二個角色升級，所以三個角色的等級和等級分別如下：

- 第一個角色 1 等，戰力為 $1 \cdot 1 + 2 = 3$
- 第二個角色 11 等，戰力為 $3 \cdot 11 + 4 = 37$
- 第三個角色 1 等，戰力為 $2 \cdot 1 + 3 = 5$

總戰力為 $3 + 37 + 5 = 45$。

2_War Power Analysis

(2 Points /8 Points)

Time Limit: 1 second
Memory Limit: 256 MB

Description

The phrase "富國強兵" (Fùguó qiángbīng) originates from an ancient Chinese anecdote recorded in the historical text "左傳" (Zuo Zhuan). This anecdote took place during the Warring States period when the ruler of the State of Jin, Duke Wen of Jin, sought to implement domestic reforms to strengthen the country's power. He asked his advisers for strategies, saying, "I desire to make the state prosperous. How can I achieve it?" One of the advisers replied, "The way to make the state prosperous is to first strengthen the military." This phrase, "富國強兵," translates to "enrich the country and strengthen the military" and has since become a common saying in China.

The meaning behind this phrase is that for a country to prosper and be strong, it must first establish a powerful military. Only with sufficient military strength can a country protect its territory, safeguard its interests, and gain advantages in diplomacy and warfare.

Now, you have the significant responsibility of managing an army consisting of n characters. The power of the i-th character is given by $a_i \cdot x + b_i$, where x represents the character's level (initially set at level 1), and each level-up requires c_i gold coins. You have a total of w gold coins. Your task is to determine the maximum power index that can be achieved for the army.

Input Format

The first line contains two positive integers n and w, representing the number of characters in the army and the amount of gold coins you possess, respectively.

The following n lines each contain three positive integers a_i, b_i, and c_i as described in the problem statement.

Output Format

Output a single positive integer, representing the maximum power of the army.

Constraints

- $1 \leq n \leq 1000$.
- $1 \leq w \leq 10^5$.
- $1 \leq a_i, b_i, c_i \leq 1000$ ($1 \leq i \leq n$).

Subtask

- Subtask 1, $n \leq 10$.
- Subtask 2, No additional constraints.

Test Cases

Input 1

```
3 10
1 2 3
3 4 1
2 3 1
```

Output 1

```
45
```

Illustrations

In Example 1, by allocating all the gold coins to the second character for upgrades, the levels and power indices of the three characters are as follows:

- The first character is at level 1, with a power index of $1 \cdot 1 + 2 = 3$.
- The second character is at level 11, with a power index of $3 \cdot 11 + 4 = 37$.
- The third character is at level 1, with a power index of $2 \cdot 1 + 3 = 5$.

The total power is $3 + 37 + 5 = 45$.

3_均值定理 (Mean Value Theorem)

(10分)

時間限制: 2 seconds
記憶體限制: 256 MB

題目敘述

小 Y 正在考微積分期末考，考卷上有一題是「請敘述並證明均值定理」，然而沒有讀書的小 Y 根本就不知道什麼是均值定理，他只好亂猜均值定理是什麼。

「對於一個由 1 和 -1 構成的序列 $a_1, a_2, \ldots, a_N$ 以及兩個整數 l, r，存在一個該序列的環狀位移（cyclic shift），滿足它的所有前綴和都在 $[l, r]$ 之間若且唯若……」

他在考卷上寫下這句話後，就寫不下去了。雖然他理所當然的沒有在這題獲得任何分數，但他在考試後還是很好奇這個問題。為了幫助他找規律，給你一個給定的序列 $a_1, a_2, \ldots, a_N$，請你對於所有的 $1 \leq k \leq N$，告訴小 Y 這個序列從位置 k 開始的環狀位移的所有前綴和之中，最小以及最大值是多少。

一個序列 $s_1, s_2, \ldots, s_N$ 的從位置 k 開始的環狀位移為序列 $s_k, s_{k+1}, \ldots, s_N, s_1, s_2, \ldots, s_{k-1}$。

一個序列 $s_1, s_2, \ldots, s_N$ 的所有前綴和為 $\{\sum_{t=1}^{i} s_t \mid 0 \leq i \leq N\}$，其中 $\sum_{t=1}^{0} s_t = 0$。

輸入格式

第一行有三個整數 N，表示序列的長度。

第二行有 N 個整數 $a_1, a_2, \ldots, a_N$，表示給定的序列。

輸出格式

輸出 N 行，其中第 k 行輸出兩個整數，分別表示從位置 k 開始的環狀位移中，前綴和的最小值與最大值。

資料範圍

- $1 \leq N \leq 10^6$
- $a_i \in \{-1, 1\}$

測試範例

輸入範例 1

```
5
1 1 -1 -1 1
```

輸出範例 1

```
0 2
-1 1
-2 1
-1 2
0 3
```

輸入範例 2

```
3
1 1 1
```

輸出範例 2

```
0 3
0 3
0 3
```

輸入範例 3

```
4
1 -1 1 -1
```

輸出範例 3

```
0 1
-1 0
0 1
-1 0
```

範例說明

範例 1 中，序列 $1, 1, -1, -1, 1$ 的所有環狀位移為：

- $k = 1$: $1, 1, -1, -1, 1$，前綴和是 $0, 1, 2, 1, 0, 1$。
- $k = 2$: $1, -1, -1, 1, 1$，前綴和是 $0, 1, 0, -1, 0, 1$。
- $k = 3$: $-1, -1, 1, 1, 1$，前綴和是 $0, -1, -2, -1, 0, 1$。
- $k = 4$: $-1, 1, 1, 1, -1$，前綴和是 $0, -1, 0, 1, 2, 1$。
- $k = 5$: $1, 1, 1, -1, -1$，前綴和是 $0, 1, 2, 3, 2, 1$。

3_Mean Value Theorem

(10 points)

Time Limit: 2 seconds
Memory Limit: 256 MB

Statement

Little Y is taking the final exam for calculus, and there is a question on the exam that says, "Please describe and prove the Mean Value Theorem." However, Little Y, who hasn't studied at all, has no idea what the Mean Value Theorem is. He can only guess what it might be.

"For a sequence $a_1, a_2, \ldots, a_N$ consisting of 1 and -1, there exists a cyclic shift of the sequence such that all of its prefix sums are within the interval $[l, r]$ if and only if..."

After writing down this sentence on the exam paper, he couldn't continue. Although he obviously didn't score any points on this question, he was still curious about it after the exam. To help him find patterns, you are given a sequence $a_1, a_2, \ldots, a_N$. For all $1 \leq k \leq N$, please tell Little Y the minimum and maximum values among the prefix sums of the cyclic shift of the sequence starting from position k.

A cyclic shift of a sequence $s_1, s_2, \ldots, s_N$ starting from position k is the sequence $s_k, s_{k+1}, \ldots, s_N, s_1, s_2, \ldots, s_{k-1}$.

The prefix sum of a sequence $s_1, s_2, \ldots, s_N$ is defined as $\{\sum_{t=1}^{i} s_t \mid 0 \leq i \leq N\}$, where $\sum_{t=1}^{0} s_t = 0$.

Input Format

The first line contains three integers, N, representing the length of the sequence.

The second line contains N integers, $a_1, a_2, \ldots, a_N$, representing the given sequence.

Output Format

Output N lines. On the k-th line, output two integers, representing the minimum and maximum values among the prefix sums of the cyclic shifts of the sequence starting from position k.

Constraints

- $1 \leq N \leq 10^6$
- $a_i \in \{-1, 1\}$

Test Cases

Input 1

```
5
1 1 -1 -1 1
```

Output 1

```
0 2
-1 1
-2 1
-1 2
0 3
```

Input 2

```
3
1 1 1
```

Output 2

```
0 3
0 3
0 3
```

Input 3

```
4
1 -1 1 -1
```

Output 3

```
0 1
-1 0
0 1
-1 0
```

Illustrations

In Example 1, all cyclic shifts of the sequence $1, 1, -1, -1, 1$ are:

- $k = 1$: $1, 1, -1, -1, 1$, whose prefix sums are $0, 1, 2, 1, 0, 1$.
- $k = 2$: $1, -1, -1, 1, 1$, whose prefix sums are $0, 1, 0, -1, 0, 1$.
- $k = 3$: $-1, -1, 1, 1, 1$, whose prefix sums are $0, -1, -2, -1, 0, 1$.
- $k = 4$: $-1, 1, 1, 1, -1$, whose prefix sums are $0, -1, 0, 1, 2, 1$.
- $k = 5$: $1, 1, 1, -1, -1$, whose prefix sums are $0, 1, 2, 3, 2, 1$.

4_多項式的根 (Roots of Polynomial)

(10 分)

時間限制: 1 second
記憶體限制: 256 MB

題目敘述

對於一個多項式 $f(x)$，我們說一個**複數** r 為 f 的根若且唯若 $f(r)=0$。

Fysty 手上有一個多項式 $P(x)$，這個多項式很神奇地能因式分解成 N 個整係數多項式 $Q_1(x),Q_2(x),\ldots,Q_N(x)$ 的乘積，即 $P(x)=Q_1(x)Q_2(x)\cdots Q_N(x)$，其中 $Q_i(x)=a_ix^2+b_ix+c_i$。

Fysty 想知道 P 有幾個**相異**的根，即有多少相異的**複數** r 使得 $P(r)=0$，但他數學不太好，因此他請你幫忙算。

輸入格式

第一行輸入一個正整數 T，代表子測試資料個數。

對於每一筆子測試資料：

第一行輸入一個正整數 N。

接下來輸入 N 行，其中第 i 行輸入三個整數 a_i,b_i,c_i，代表 $Q_i(x)=a_ix^2+b_ix+c_i$。

注意到 $Q_i(x)$ 可以是一次式或是只有常數項，唯獨保證不會是零多項式。

輸出格式

每一筆子測試資料輸出一行，這行只有一個整數 cnt，代表 P 有 cnt 個相異的根。

資料範圍

- $1\le T\le 10^4$
- $1\le N\le 2\cdot 10^5$
- $|a_i|,|b_i|,|c_i|\le 10^9$
- a_i,b_i,c_i 不全為 0
- 保證 N 的總和不超過 $2\cdot 10^5$

測試範例

輸入範例 1

```
4
4
1 -1 1
1 2 0
```

```
1 -2 1
-50 100 0
5
0 0 45510
0 45 510
4 55 10
45 5 10
455 1 0
3
0 1 -2
0 2 -3
0 4 -6
2
0 0 1
0 0 -1
```

輸出範例 1

```
6
7
2
0
```

範例說明

令 $i = \sqrt{-1}$。

第一筆子測試資料中，P 的根有 $0, 1, 2, -2, \frac{1+i\sqrt{3}}{2}, \frac{1-i\sqrt{3}}{2}$。

第二筆子測試資料中，P 的根有 $0, \frac{-35}{3}, \frac{-1}{455}, \frac{-55+\sqrt{2865}}{8}, \frac{-55-\sqrt{2865}}{8}, \frac{-1+i\sqrt{71}}{18}, \frac{-1-i\sqrt{71}}{18}$。

第三筆子測試資料中，P 的根有 $2, \frac{3}{2}$。

第四筆子測試資料中，P 沒有任何根。

4_Roots of Polynomial

(10 points)

Time Limit: 1 second
Memory Limit: 256MB

Statement

For a polynomial $f(x)$, we say a **complex** number r is a root of f if and only if $f(r) = 0$.

Fysty has a polynomial $P(x)$ in his hands. This polynomial can miraculously be decomposed into N polynomials with integer coefficients $Q_1(x), Q_2(x), \ldots, Q_N(x)$. Basically $P(x) = Q_1(x)Q_2(x)\cdots Q_N(x)$, where $Q_i(x) = a_i x^2 + b_i x + c_i$.

Fysty wants to know how many distinct roots P has. In other words, how many different **complex** number r are there such that $P(r) = 0$. But he is terrible at math, so he begs for your help.

Input Format

The first line of input contains a positive integer T, representing the number of testcases.

For each testcase:

The first line contains a positive integer N.

Then there are N lines. The i-th line contains three integers a_i, b_i, c_i, representing $Q_i(x) = a_i x^2 + b_i x + c_i$.

Note that the degree of $Q_i(x)$ can be 1 or 0, but it will never be a zero polynomial

Output Format

For each testcase, output one line. This line contains one integer cnt, meaning there are cnt distinct roots of P.

Constraints

- $1 \le T \le 10^4$
- $1 \le N \le 2 \cdot 10^5$
- $|a_i|, |b_i|, |c_i| \le 10^9$
- a_i, b_i, c_i are not all 0
- It is guaranteed that the sum of N over all testcases does not exceed $2 \cdot 10^5$

Test Cases

Input 1

```
4
4
1 -1 1
1 2 0
1 -2 1
-50 100 0
5
0 0 45510
0 45 510
4 55 10
45 5 10
455 1 0
3
0 1 -2
0 2 -3
0 4 -6
2
0 0 1
0 0 -1
```

Output 1

```
6
7
2
0
```

Illustrations

Let $i = \sqrt{-1}$.

In the first testcase, the roots of P are $0, 1, 2, -2, \frac{1+i\sqrt{3}}{2}, \frac{1-i\sqrt{3}}{2}$.

In the second testcase, the roots of P are $0, \frac{-35}{3}, \frac{-1}{455}, \frac{-55+\sqrt{2865}}{8}, \frac{-55-\sqrt{2865}}{8}, \frac{-1+i\sqrt{71}}{18}, \frac{-1-i\sqrt{71}}{18}$.

In the third testcase, the roots of P are $2, \frac{3}{2}$。

In the fourth testcase, P has no roots.

5_When2meet

(3分/3分/9分)
時間限制: 2 seconds
記憶體限制: 512 MB

題目敘述

在一個沒有作業沒有 deadline 也沒有考試的世界裡，有一群很喜歡出去玩的人，他們平時的興趣就是認識不同的朋友，並且約大家一起出去玩。

在這個世界的初始設定中，每個人都是互相不認識的邊緣人，自己活在一個人的社交圈中。接下來會有 Q 個事件依序發生：

- 有兩個人互相認識了，這兩個人的社交圈從此將會合併成為一個比較大的社交圈
- 有 k_i 個人想要約出去玩，這群人能約出去玩的先決條件是他們都屬於同一個社交圈，也就是說，如果這 k_i 個人之中有任何一個人在約出去玩的當下跟其他人分屬不同的社交圈子，那這個出遊計畫肯定會流局。

你以為問題就這麼簡單嗎？當然不可能。在剛剛的第二種事件中，假設那 k_i 個人真的約成了，他們很開心的出遊並享受現充生活，這時，團裡的其中一個人提出了問題：「我們這團人是從什麼時候開始認識的呀？」，對，這才是希望你幫忙解決的問題。對於每一組成功約出去的 k_i 個人，想請你告訴他們，他們是從什麼時間點開始變成同一個社交圈的。

輸入格式

第一行會有兩個正整數 N, Q，分別代表總共有多少人，以及接下來有多少事件。

接下來的 Q 行每行會先有一個數字 $e_i \in \{1, 2\}$ 代表事件類型：

- 若 $e_i = 1$，接下來會有兩個數字 a_i, b_i，代表編號為 a_i 的人和編號為 b_i 的人在時間點 i 時互相認識。
- 若 $e_i = 2$，接著會先有一個數字 k_i，代表現在希望能約 k_i 個人一起出去玩，接下來會有 k_i 個數字 $x_1, x_2, \cdots, x_{k_i}$，代表我們希望能約一起出去玩的人的編號。

輸出格式

對於每個 $e_i = 2$, 請輸出一個數值代表這 k_i 個人是從什麼時間開始變成同一個社交圈的，若這 k_i 個人在要約出去玩的當下還不在同一個社交圈裡，則輸出 -1。

資料範圍

- $1 \leq N \leq 2 \cdot 10^5$
- $1 \leq Q \leq 5 \cdot 10^5$
- $1 \leq k_i \leq N, \sum k_i \leq 5 \cdot 10^5$
- $1 \leq a_i, b_i, x_i \leq N$

子任務

- 子任務 1 滿足 $k_i = 2$

- 子任務 2 滿足 $N \leq 1000, Q \leq 5000, \sum k_i \leq 5000$
- 子任務 3 無其他限制

測試範例

輸入範例 1

```
5 10
1 1 2
1 2 3
2 2 1 3
2 2 1 5
1 3 4
1 2 5
2 2 2 5
1 1 3
1 2 4
2 2 1 4
```

輸出範例 1

```
2
-1
6
5
```

輸入範例 2

```
5 10
1 1 2
1 2 3
2 3 1 2 3
2 3 1 2 4
1 3 4
1 2 5
2 3 2 3 4
1 1 3
1 2 4
2 3 1 4 5
```

輸出範例 2

```
2
-1
5
6
```

範例說明

在範例測資 1 中：

- 1, 3 兩個人在時間點 2 時藉由共同好友互相認識，因此輸出 2
- 1, 5 兩個人在詢問時間當下還互相不認識對方，因此輸出 -1
- 2, 5 兩個人在時間點 6 時互相認識，因此輸出 6
- 1, 4 兩個人在時間點 5 時藉由兩層共同好友互相認識，因此輸出 5

5_When2meet

(3 Points/3 Points/9 Points)

Time Limit: 2 seconds
Memory Limit: 512 MB

Statement

In a world without assignments, deadlines, or exams, there is a group of people who love going out and having fun. Their main interest is getting to know different friends and organizing outings together.

In the initial setup of this world, each person is a stranger to everyone else, living in their own social circle. There will be a series of Q events happening:

- Two people become acquainted with each other, and their social circles merge into a larger one.
- A group of k_i people want to plan an outing together. The prerequisite for this group is that they all belong to the same social circle. In other words, if any person in this group belongs to a different social circle at the time of the outing, the plan will be unsuccessful.

But is the problem really that simple? Of course not. In the second type of event mentioned earlier, suppose the group of k_i people successfully plans the outing and enjoys their time together. Then, one person in the group raises a question: "When did we all start knowing each other?" Yes, this is the question they want you to help solve. For each group of k_i people who successfully plan their outing, they want to know at what time point they became part of the same social circle.

Input Format

The first line will contain two positive integers, N and Q, representing the total number of people and the number of events to follow.

The next Q lines each start with a number $e_i \in 1, 2$, representing the type of event:

- If $e_i = 1$, there will be two numbers a_i and b_i, indicating that person with ID a_i and person with ID b_i became acquainted with each other at time point i.
- If $e_i = 2$, there will be a number k_i, representing the desired number of people to gather for an outing. Following that, there will be k_i numbers $x_1, x_2, \cdots, x_{k_i}$, representing the IDs of the people we wish to invite for the outing.

Output Format

For each $e_i = 2$, please output a value representing the time when the group of k_i people became part of the same social circle. If the k_i individuals are not in the same social circle at the time of the outing plan, output -1.

Constraints

- $1 \leq N \leq 2 \cdot 10^5$
- $1 \leq Q \leq 5 \cdot 10^5$

- $1 \leq k_i \leq N, \sum k_i \leq 5 \cdot 10^5$
- $1 \leq a_i, b_i, x_i \leq N$

Subtasks

- Subtask 1 satisfies $k_i = 2$.
- Subtask 2 satisfies $N \leq 1000$, $Q \leq 5000$, and $\sum k_i \leq 5000$.
- Subtask 3 has no additional limits.

Test Cases

Input 1

```
5 10
1 1 2
1 2 3
2 2 1 3
2 2 1 5
1 3 4
1 2 5
2 2 2 5
1 1 3
1 2 4
2 2 1 4
```

Output 1

```
2
-1
6
5
```

Input 2

```
5 10
1 1 2
1 2 3
2 3 1 2 3
2 3 1 2 4
1 3 4
1 2 5
2 3 2 3 4
1 1 3
1 2 4
2 3 1 4 5
```

Output 2

```
2
-1
5
6
```

Illustrations

In Example Test Case 1:

- Two people, 1 and 3, become part of the same social circle through a mutual friend at time point 2. Therefore, the output is 2.
- Two people, 1 and 5, do not belong to the same social circle at the time of the query. Therefore, the output is -1.
- Two people, 2 and 5, become part of the same social circle at time point 6. Therefore, the output is 6.
- Two people, 1 and 4, become part of the same social circle through two layers of mutual friends at time point 5. Therefore, the output is 5.

6_跑馬燈 (Marquee)

(2分/4分/14分)

時間限制: 1 second
記憶體限制: 512 MB

題目敘述

將一生奉獻給回文的師老師總是沒日沒夜的埋首於回文字串的研究，師老師的研究筆記上是這樣寫的：「一個回文字串就是正著念與倒著念都一樣的字串，舉例來說：『師老師』就是一個回文字串」。然而、龐大的工作量與長期的睡眠缺乏終於讓師老師病倒了，僅存一絲氣息的師老師看見了人生跑馬燈。

師老師的人生跑馬燈反覆顯示著字串 S，也就是說，字串中的最後一個字後面會接著字串中的第一個字、週而復始。然而由於跑馬燈的寬度只有 N，只能顯示 S 中的一部份。舉例來說，如果跑馬燈以 s_i 作為開頭，則跑馬燈顯示的字串 T_i 就會是

$$T_i = s_i s_{(i+1)\%|S|} \dots s_{(i+N-1)\%|S|} \tag{1}$$

師老師看到跑馬燈之後對回文產生了新的疑問：「從 S 的不同位置開頭顯示的字串各自究竟有幾個回文子字串呢？」正式一點來說，就是對每個從 i 開頭的字串 $T_i = t_{i,0}t_{i,1}\dots t_{i,N-1}$，我們要計算有幾組不同的整數數對 (ℓ, r) 滿足 $0 \le \ell \le r \le N-1$ 使得

$$t_{i,\ell}t_{i,\ell+1}\dots t_{i,r} \tag{2}$$

是一個回文字串。醫生認為只要解答了這個問題，師老師很可能可以使用意志力來戰勝病魔。為了拯救師老師，請告訴他這個問題的答案吧！

輸入格式

輸入的第一行包含一個正整數 N，代表跑馬燈的寬度。

第二行則包含一個字串 S，代表跑馬燈顯示的文字。

輸出格式

輸出 $|S|$ 行，每行輸出一個整數，其中第 i 行的輸出代表以 s_i 開頭的跑馬燈顯示的字串 t_i 中的回文子字串個數。

資料範圍

- $1 \le |S| \le 100\,000$。
- $1 \le N \le |S|$。
- 字串 S 中只包含小寫英文字母。

子任務

- 子任務 1, $|S| \le 200$。
- 子任務 2, $|S| \le 5000$。
- 子任務 3, 無額外限制。

測試範例

輸入範例 1

```
5
abcbaa
```

輸出範例 1

```
7
7
8
9
8
7
```

輸入範例 2

```
7
cababac
```

輸出範例 2

```
12
12
11
11
11
11
12
```

範例說明

在範例 1 中，跑馬燈顯示的六種樣式分別為：

- $T_0 = \texttt{abcba}$：有 7 個不同的回文子字串，除了五組 (p, p) 之外的還有 $(0, 4)$ 以及 $(1, 3)$。
- $T_1 = \texttt{bcbaa}$：有 7 個不同的回文子字串，除了五組 (p, p) 之外的還有 $(0, 2)$ 以及 $(3, 4)$。
- $T_2 = \texttt{cbaaa}$：有 8 個不同的回文子字串。
- $T_3 = \texttt{baaab}$：有 9 個不同的回文子字串。
- $T_4 = \texttt{aaabc}$：有 8 個不同的回文子字串。
- $T_5 = \texttt{aabcb}$：有 7 個不同的回文子字串。

6_Marquee

(2 points/4 points/14 points)

Time Limit: 1 second
Memory Limit: 512 MB

Description

Dr.D dedicated his entire life to the study of palindromes, spending countless days and nights buried in research on palindrome strings. In his research notes, Dr.D wrote, "A palindrome string is a string that reads the same forward and backward. For example, 'DrD' is a palindrome string." However, the massive workload and lack of sleep eventually caused Dr.D to fall ill, and with only a breath left, he saw the marquee of his life.

The marquee of his life repeatedly displays a string S, starting from the beginning each time the end is reached. However, due to the marquee's width being limited to N, only a part of S can be displayed. For instance, if the marquee starts with s_i, then the displayed string T_i would be

$$T_i = s_i s_{(i+1)\%|S|} \dots s_{(i+N-1)\%|S|} \tag{3}$$

Upon seeing the marquee, Dr.D developed a new question about palindromes: "How many palindromic substrings are there for each string that is displayed by starting from different positions in S?" Formally, for each start point i, we can find a string $T_i = t_{i,0}t_{i,1}\dots t_{i,N-1}$, and we should find the number of distinct pairs of integers (ℓ, r) satisfying $0 \le \ell \le r \le N-1$ such that

$$t_{i,\ell}t_{i,\ell+1}\dots t_{i,r} \tag{4}$$

is a palindrome. The doctor believed that if Dr.D figure out the answer to this question, he could potentially overcome his illness with willpower. Please help to save Dr.D by providing the answer to this question!

Input Format

The input contains two lines.

The first line contains a single integer N, representing the width of the marquee.

The second line contains a string S, representing the text displayed on the marquee.

Output Format

Output $|S|$ lines, each line containing an integer. The output on the i-th line represents the number of palindrome substrings in the string T_i displayed on the marquee, where T_i starts with s_i.

Constraints

- $1 \le |S| \le 100\,000$.
- $1 \le N \le |S|$.
- The string S contains only lowercase Latin letters.

Subtasks

- Subtask 1, $|S| \leq 200$.
- Subtask 2, $|S| \leq 5000$.
- Subtask 3, No additional constraints.

Test Cases

Input 1

```
5
abcbaa
```

Output 1

```
7
7
8
9
8
7
```

Input 2

```
7
cababac
```

Output 2

```
12
12
11
11
11
11
12
```

Illustrations

In example 1, all T_i displayed on marquee are:

- $T_0 =$ `abcba`: There are 7 distinct palindrome substrings, including five pairs of (p, p), as well as $(0, 4)$ and $(1, 3)$ as additional pairs.

- $T_1 =$ `bcbaa`: There are 7 distinct palindrome substrings, including five pairs of (p, p), as well as $(0, 2)$ and $(3, 4)$ as additional pairs.
- $T_2 =$ `cbaaa`: There are 8 distinct palindrome substrings.
- $T_3 =$ `baaab`: There are 9 distinct palindrome substrings.
- $T_4 =$ `aaabc`: There are 8 distinct palindrome substrings.
- $T_5 =$ `aabcb`: There are 7 distinct palindrome substrings.

7_召喚到異世界 (Summoned to Another World)

(8分 /17分)

時間限制: 3 seconds
記憶體限制: 256 MB

題目敘述

YTP 國是異世界的一個國家，它由 N 個島嶼和 M 座雙向大橋組成，每座橋會連接兩個不同的島嶼，注意兩個島嶼之間可能有不只一座大橋。

最近 YTP 國受到魔王軍的騷擾，對於所有 $1 \leq i \leq M$，魔王在第 i 座大橋上放置了一隻等級 l_i 的魔物，這種魔物就算死了，一到隔天就會復活。不只如此，在接下來 Q 天中，第 i 天會有一位魔王軍幹部在島嶼 y_i 發動襲擊。

YTP 國為了應對魔王軍的襲擊，接下來 Q 天的每一天都會召喚一位勇者。每次召喚的勇者的等級可以是任意的**非負整數**，而且每一位勇者都會獲得一個神聖道具，這個道具可以在一次戰鬥中，將勇者自身的等級暫時變成無限大，但戰鬥完就變回原本的等級，並且僅能使用一次。不過召喚魔法並不完善，所以有諸多缺點，其中最重要的就是第 i 位勇者一定會被召喚到島嶼 x_i 上。

勇者僅能通過大橋以抵達相鄰的島嶼，而想要通過一座大橋就必須擊敗橋上的魔物。想成功擊敗一隻等級 k 的魔物，勇者自身的等級必須至少是 k。

因為召喚的勇者等級越高越費魔力，所以 YTP 國希望每次召喚的勇者等級越低越好。請幫 YTP 國計算出第 i 位勇者等級至少需要是多少才能成功抵達 y_i，或是告訴 YTP 國此勇者無論如何都無法抵達島嶼 y_i。

註：你可以假設勇者們戰鬥以及通過橋和島嶼所需的時間趨近於 0，而且每一位勇者在當天結束時就會回到他原本的世界，還有，你不用考慮勇者的等級能否擊敗魔王軍幹部。

輸入格式

第一行輸入三個正整數 N, M, Q。

接下來輸入 M 行，第 i 行輸入三個整數 u_i, v_i, l_i，代表第 i 座大橋連接島嶼 u_i, v_i，而且橋上有一隻等級 l_i 的魔物。

接下來輸入 Q 行，第 i 行輸入兩個正整數 x_i, y_i，代表第 i 位勇者一定會被召喚到的島嶼 x_i，以及魔王軍幹部會在第 i 天襲擊島嶼 y_i。

輸出格式

輸出 Q 行，第 i 行輸出一個整數 k_i，如果第 i 位勇者無論如何都無法抵達 y_i，則 $k_i = -1$，否則 k_i 代表第 i 位勇者的等級至少需要是 k_i。

資料範圍

- $2 \leq N \leq 10^5$
- $1 \leq M \leq 10^5$
- $1 \leq Q \leq 10^5$
- $1 \leq u_i, v_i, x_i, y_i \leq N$

- $0 \leq l_i \leq 10^9$
- $u_i \neq v_i$
- $x_i \neq y_i$

子任務

- 子任務 1 滿足 $N, Q \leq 1000, M \leq 5000$
- 子任務 2 沒有其他限制

測試範例

輸入範例 1

```
10 10 5
1 4 1
2 3 3
1 2 2
2 5 2
5 3 2
5 4 4
6 7 8
7 8 455
8 9 10
9 10 6
1 5
3 4
6 10
7 8
1 9
```

輸出範例 1

```
1
2
10
0
-1
```

範例說明

第一位勇者可以依序走第 $1, 6$ 座大橋，並選擇在第 6 座大橋使用神聖道具。

第二位勇者可以依序走第 $5, 6$ 座大橋，並選擇在第 6 座大橋使用神聖道具。

第三位勇者可以依序走第 $7, 8, 9, 10$ 座大橋，並選擇在第 8 座大橋使用神聖道具。

第四位勇者可以依序走第 8 座大橋，並選擇在第 8 座大橋使用神聖道具，注意到等級一定要是非負整數。

第五位勇者無論如何都走不到目的地。

7_Summoned to Another World

(8 points /17 points)

Time Limit: 3 seconds
Memory Limit: 256MB

Statement

YTP is a country in another world, consisting of N islands and M bidirectional bridges. Each bridge connects two different islands. There may be more than one bridge between two islands.

Recently, YTP has been harassed by the Demon King's army. The Demon King has placed a level l_i monster on the i-th bridge for all $1 \leq i \leq M$. These monsters, even if killed, will respawn at the beginning of the next day. Furthermore, a high-ranking officer of the Demon King's army will launch an attack on island y_i on the i-th day of the next Q days.

To counter the attacks from the Demon King's army, YTP will summon a hero each day for the next Q days. The summoned hero's level can be any **non-negative integer**, and each hero will receive a holy item. This item can make the hero's level infinite for one battle. But after the battle, his/her level returns back to the original value. The holy item can only be used once for each hero. However, the summoning magic isn't perfect. It has many flaws. The most important flaw of them all is that the i-th hero will have to be summoned to island x_i.

Heroes can only cross bridges to reach adjacent islands. To cross a bridge, they must defeat the monster on the bridge. To successfully defeat a monster with level k, the hero's level must be at least k.

Since summoning heroes with higher levels consumes more magic power, YTP wants to summon heroes with the lowest possible levels each time. Please help YTP Country calculate the minimum level needed for the i-th hero to successfully reach y_i on the i-th day. Also, if it is impossible for the hero to reach island y_i regardless of their level, report this issue to YTP.

Note: You can assume that the time it takes to battle monsters, cross bridges, and travel through islands is practically 0. Also, each hero will return to their original world at the end of the i-th day. You do not need to consider whether the heroes can defeat the high-ranking officers of the Demon King's army.

Input Format

The first line of input contains three positive integers N, M, Q.

The next M lines contain three integers u_i, v_i, l_i. The i-th bridge connects islands u_i, v_i, and there is a monster with level l_i on the bridge.

The next Q lines contain two integers x_i, y_i. The i-th hero will be summoned to island x_i, and a high-ranking officer of the Demon King's army will attack island y_i on the i-th day.

Output Format

Output Q lines. The i-th line should contain an integer k_i. If the i-th hero can't reach y_i regardless of their level, then $k_i = -1$. Otherwise, k_i represents the minimum level needed for the i-th hero.

Constraints

- $2 \leq N \leq 10^5$
- $1 \leq M \leq 10^5$
- $1 \leq Q \leq 10^5$
- $1 \leq u_i, v_i, x_i, y_i \leq N$
- $0 \leq l_i \leq 10^9$
- $u_i \neq v_i$
- $x_i \neq y_i$

Subtasks

- Subtask 1 satisfies that $N, Q \leq 1000, M \leq 5000$. (8 points)
- Subtask 2 has no additional constraint. (17 points)

Test Cases

Input 1

```
10 10 5
1 4 1
2 3 3
1 2 2
2 5 2
5 3 2
5 4 4
6 7 8
7 8 455
8 9 10
9 10 6
1 5
3 4
6 10
7 8
1 9
```

Output 1

```
1
2
10
0
-1
```

Illustrations

The first hero can go through bridges $1, 6$ and choose to use the holy item on bridge 6.

The second hero can go through bridges $5, 6$ and choose to use the holy item on bridge 6.

The third hero can go through bridges $7, 8, 9, 10$ and choose to use the holy item on bridge 8.

The fourth hero can go through bridge 8 and choose to use the holy item on bridge 8. Note that the hero's level has to be a non-negative integer.

The fifth hero can never reach his desired destination.