

## Задача А. Як розрізняти генератори

Назва вхідного файлу: `standard input`  
Назва вихідного файлу: `standard output`  
Ліміт часу: `0.6 seconds`  
Ліміт використання пам'яті: `256 megabytes`

Поліна хоче зробити своєму хлопцю подарунок. Оскільки це має бути щось особливе, вона вирішила подарувати справжній випадковий масив. Але ніхто не може зробити випадковий масив власноруч, тому їй довелося придбати генератор випадкових масивів. А оскільки в цей час у магазині відбувся розпродаж, то Поліна, як справжня дівчина, не могла повернутися додому з тим, що їй було потрібно спочатку.

Таким чином вона придбала генератор, що вміє не тільки генерувати випадкові масиви, а й випадкові перестановки (а коштує стільки ж!).

Детальніше, генерація випадкового масиву полягає в тому, що рівномірно обирається один з можливих масивів довжини  $N$ , що містить цілі числа від 1 до  $N$ .

У свою чергу, генерація випадкової перестановки полягає в тому, що рівномірно обирається одна з можливих перестановок довжини  $N$ .

Проте, генератор виявився бракованим, через що він видає то випадкові перестановки, то випадкові масиви.

Поліна вже використала генератор  $K$  разів. Вона просить вас, як її кращого друга, визначити, які із згенерованих наборів були отримані в режимі генерації перестановки, а які - в режимі генерації масиву.

Але, оскільки набори можуть бути досить великі, вона може сказати вам не більше, ніж 30000 чисел загалом з усіх наборів (детальніше див. розділ "Протокол взаємодії").

Відповідь на тест буде вважатися правильною, якщо вона містить не більше двох неправильних припущень щодо режиму генерації. Гарантується, що кількість тестів не перевищує 100.

### Протокол взаємодії

У першому рядку вхідних даних подаються два числа -  $N$  ( $30000 \leq N \leq 40000$ ) та  $K$  ( $30 \leq K \leq 40$ ).

Після того, як ці дані були зчитані відбувається інтерактивна взаємодія. Ви можете використовувати такі запити:

- ?  $t x$  - дізнатись елемент з номером  $x$  ( $1 \leq x \leq N$ ) у наборі з номером  $t$  ( $1 \leq t \leq K$ ). У відповідь ви отримаєте одне ціле число - значення елементу з номером  $x$  у наборі з номером  $t$ .
- !  $a_1 a_2 \dots a_K$  - надати відповідь на задачу -  $N$  чисел, розділених пробілом. Тут  $a_i$  має дорівнювати 0, якщо ви вважаєте, що набір з номером  $i$  був створений у режимі генерації масиву, та 1 - якщо у режимі генерації перестановки. Після цього запиту ваша програма має припинити свою роботу.

Максимальна дозволена кількість запитів першого типу - 30000.

### Приклад

standard input	standard output
4 2	? 1 1
1	? 1 2
3	? 1 3
4	? 1 4
2	? 2 1
1	? 2 2
2	? 2 3
3	? 2 4
4	! 1 0

## Зауваження

Як бачимо, у тестовому прикладі обидва набори виявились рівними [1; 2; 3; 4] та [1; 3; 4; 2], що могли бути отримані як в режимі генерації перестановки, так і в режимі генерації масиву. Але, оскільки дві помилки у відповіді допускаються, то незалежно від того, як будуть визначені режими генерації цих наборів, цей тестовий приклад буде зарахований.

Наведений тестовий приклад не відповідає обмеженням задачі і наданий лише для пояснення формату взаємодії. Цей тест **не** включений до набору тестів.

## Задача В. Брати чи не брати?

Назва вхідного файлу:	standard input
Назва вихідного файлу:	standard output
Ліміт часу:	2 seconds
Ліміт використання пам'яті:	256 megabytes

Шопінг - складна справа. Особливо, коли намагаєшся серед всіх речей обрати одну, і вона має бути найкращою. Але Поліна не впадає у відчай, бо знає, що ви їй допоможете!

Всього Поліна планує відвідати  $K$  магазинів. В кожному з них вона планує переглянути  $N$  речей. Кожна з цих речей має деякий умовний параметр  $a_i$  - чим більше цей параметр, тим більше річ пасує Поліні. Необхідно з усіх речей в межах кожного магазину обрати одну з найбільшим значенням цього параметра (гарантується, що в межах магазину усі  $a_i$  різні).

Найбільша складність полягає в тому, що Поліна передивляється речі в деякому **випадковому** порядку, і після того, як вона подивилась чергову річ, вона має одразу сказати - бере вона її чи ні. Якщо ні, то вона переходить до наступної речі, а якщо так, то забирає поточну річ і залишає поточний магазин. При цьому Поліна має взяти **рівно** одну річ в кожному магазині.

Зрозуміло, що в деяких випадках Поліна зможе вгадати найкращу річ, а в деяких - ні. Ваша задача - допомогти їй знайти найкращу річ принаймні у 25% магазинів. Гарантується, що кількість тестів не перевищує 100.

### Протокол взаємодії

У першому рядку вхідних даних подаються два числа -  $K$  ( $500 \leq K \leq 1000$ ) та  $N$  ( $100 \leq N \leq 200$ ). При цьому  $N \cdot K \leq 100000$ .

Після того, як ці дані були зчитані відбувається інтерактивна взаємодія.

Для кожного з  $K$  магазинів відбувається такі кроки:

1. До вхідних даних подається одне ціле число  $a_i$  ( $1 \leq a_i \leq 10^9$ ) - параметр поточної речі.
2. Ви подаєте до вихідних даних одне з двох чисел: 0 чи 1.
3. Якщо це 1 - поточна річ обирається. Якщо це останній магазин, то ваша програма має завершити роботу, якщо ні - взаємодія починається з пункту 1 для **наступного** магазину.
4. Якщо це 0 - поточна річ відкидається. Якщо це була остання річ у поточному магазині, то це призведе до вердикту "Неправильна відповідь" (бо необхідно обрати принаймні одну річ). В іншому випадку, взаємодія продовжується знову з пункту 1 для **поточного** магазину.

Розв'язок буде зарахований, якщо принаймні 25% обраних речей будуть мати найбільший параметр  $a$  в межах магазину, в якому вони знаходяться.

### Приклад

standard input	standard output
2 5	0
50	0
60	0
70	0
80	1
100	1
20	

### Зауваження

Тут Поліна точно вгадала у першому магазині, бо обрала річ з параметром 100, що більше за параметри усіх інших речей в першому магазині.

У другому магазині, навіть якщо параметр 20 мала не найкраща річ, то тест все одно буде зарахований, адже необхідно вгадати принаймні у 25% магазинів.

Наведений тестовий приклад не відповідає обмеженням задачі і наданий лише для пояснення формату взаємодії. Цей тест **не** включений до набору тестів.

## Задача С. Вгадай максимум

Назва вхідного файлу: `standard input`  
Назва вихідного файлу: `standard output`  
Ліміт часу: `3 seconds`  
Ліміт використання пам'яті: `256 megabytes`

Поліна хоче зіграти з вами в одну цікаву гру. В неї є набір цілих чисел, а вам необхідно знайти серед них найбільше.

Але, якби ви знали ці числа, то гра не була б цікавою! Отже, ви лише можете ставити Поліні питання вигляду: "чи правда, що елемент з індексом  $i$  більше за число  $x$ ". На основі цих даних вам необхідно буде визначити найбільший елемент у наборі.

### Протокол взаємодії

У першому рядку вхідних даних подається число  $N$  ( $1 \leq N \leq 200000$ ).

Після того, як ці дані були зчитані відбувається інтерактивна взаємодія.

Ви можете використовувати два типи запитів:

- $i \ x$ , де  $1 \leq i \leq N$  та  $1 \leq x \leq 10^9$ . Це запит: "чи правда, що елемент з номером  $i$  більше за  $x$ ". У відповідь ви отримуєте 0, якщо ні, та 1, якщо так.
- $-1 \ x$ , де  $1 \leq x \leq 10^9$ . Це відповідь: " $x$  - це найбільший елемент у наборі". Після цього запиту ваша програма має завершитися.

Максимальна дозволена кількість запитів цього типу - 201000.

Гарантується, що елементи масиву визначені заздалегідь і не змінюються в залежності від запитів. Гарантується, що елементи масиву приймають значення від 1 до  $10^9$ .

### Приклад

standard input	standard output
2	1 1
0	2 1
1	2 2
0	-1 2

## Задача D. Арифметична прогресія

Назва вхідного файлу: `standard input`  
Назва вихідного файлу: `standard output`  
Ліміт часу: 0.3 seconds  
Ліміт використання пам'яті: 256 megabytes

Нещодавно Поліна знайшла набір з  $N$  цілих чисел. Вона впевнена, що якщо його впорядкувати за зростанням, то він виявиться арифметичною прогресією.

Нагадаємо, що арифметична прогресія з різницею  $d$  та перший членом  $x_1$  - це набір чисел  $x_1, x_2, \dots, x_N$  такий, що  $x_i = x_1 + (i - 1) \cdot d$ .

Поліна просить вас допомогти їй довести, що вона має рацію, тобто знайти ціле невід'ємне  $d$  та ціле  $x_1$ . Проте, надати вам весь набір було б занадто легко, тому Поліна готова відповісти на не більше ніж 60 питань одного з двох типів. Перший тип - "який елемент знаходиться на позиції  $i$ ?". Другий тип - "чи є в наборі елемент більший за  $x$ ?".

Чи зможете ви допомогти Поліні?

### Протокол взаємодії

У першому рядку вхідних даних подається ціле число  $N$  ( $2 \leq N \leq 10^6$ ).

Після того, як ці дані були зчитані відбувається інтерактивна взаємодія. Ви можете використовувати такі запити:

- `? i` - дізнатись елемент з номером  $i$  ( $1 \leq i \leq N$ ) у наборі. У відповідь ви отримаєте одне ціле число - значення елемента  $a_i$  ( $1 \leq a_i \leq 10^9$ ).
- `> x` - дізнатись, чи є в наборі елемент, більший за  $x$  ( $1 \leq x \leq 10^9$ ). У відповідь ви отримаєте одне ціле число - 0, якщо такого елемента немає, та 1, якщо є.
- `! x1 d` - надати відповідь на задачу - два цілих числа  $x_1$  ( $1 \leq x_1 \leq 10^9$ ) та  $d$  ( $0 \leq d \leq 10^9$ ) - перший член арифметичної прогресії та її різниця відповідно. Після цього запиту ваша програма має припинити свою роботу.

Максимальна дозволена кількість запитів першого та другого типу загалом - 60.

### Приклад

standard input	standard output
3	? 1
10	? 2
2	? 3
6	> 10
0	> 9
1	! 2 4

### Зауваження

Гарантується, що у кожному тесті відповідь існує і єдина.

## Задача Е. Знову арифметична прогресія

Назва вхідного файлу:	standard input
Назва вихідного файлу:	standard output
Ліміт часу:	0.75 seconds
Ліміт використання пам'яті:	256 megabytes

Поліні подобаються всілякі квести. Цього разу перед нею трапився набір різних цілих додатніх чисел, який вона має забрати собі. Але відповідно до правил квесту забрати елементи з цього набору можна лише таким чином - обрати числа  $x_0$  та  $d$  ( $d > 1$ ), та забрати всі елементи набору, що входять до арифметичної прогресії з першим членом  $x_0$  та різницею  $d$  (формально кажучи, можна забрати всі елементи  $a_i$ , що можна подати у вигляді  $a_i = x_0 + k \cdot d$ , для деякого цілого  $k$ ).

Звичайно, Поліна хоче забрати собі якомога більше чисел. Вона вже розв'язала цю задачу, але хоче перестрахуватися, тому просить вас теж її розв'язати, тобто знайти, яку найбільшу кількість чисел Поліна може забрати собі.

### Формат вхідних даних

Перший рядок містить одне ціле число  $n$  ( $1 \leq n \leq 10^5$ ) — кількість чисел у наборі.

Другий рядок містить  $n$  цілих чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) — елементи набору. Гарантується, що всі  $a_i$  попарно різні.

### Формат вихідних даних

Виведіть одне ціле число  $x$  ( $0 \leq x \leq n$ ) - найбільшу кількість чисел, яку Поліна може забрати собі.

### Приклади

standard input	standard output
5 1 2 3 4 7	3
7 1 2 10 4 7 3 13	5

### Зауваження

У першому прикладі Поліна може вибрати  $x_0 = 1, d = 2$  та забрати числа 1, 3, 7.

У другому прикладі Поліна може вибрати  $x_0 = 1, d = 3$  та забрати числа 1, 4, 7, 10, 13.

## Задача F. Все по 1000

Назва вхідного файлу: `standard input`  
Назва вихідного файлу: `standard output`  
Ліміт часу: `0.25 seconds`  
Ліміт використання пам'яті: `256 megabytes`

У Поліни є таблиця  $1000 \times 1000$ , в кожній комірці якої записане ціле число від 1 до 1000. Вона називає *шляхом* у цій таблиці послідовність комірок, що починається у лівому верхньому куті таблиці, складається із пересувань донизу та праворуч та закінчується у правому нижньому куті. Сума чисел, що записані у комірках, через які проходить шлях, називаються його довжиною.

Поліну цікавить середнє арифметичне довжин усіх шляхів в таблиці. Звичайно, вона просить вас порахувати цю величину.

Але, якби Поліна сказала вам усі числа в таблиці, то було б занадто легко. Тому ви можете запитати в неї лише деякі 1500 чисел з таблиці. Крім того, вона гарантує, що числа в таблиці були згенеровані незалежно, випадково і рівномірно на деякому (можливо різному в різних тестах) відрізьку  $[a; b]$  ( $1 \leq a \leq b \leq 1000$ ).

Якщо ваша відповідь  $X$ , а реальне значення середнього арифметичного  $Y$ , то відповідь буде вважатися правильною, якщо  $|X - Y| \leq 70 * (b - a + 1)$ .

### Протокол взаємодії

Інтерактивна взаємодія відбувається одразу.

Ви можете використовувати такі запити:

- $? i j$  - дізнатись елемент на перетині рядка  $i$  та стовпця  $j$  ( $1 \leq i \leq 1000, 1 \leq j \leq 1000$ ). У відповідь ви отримаєте одне ціле число - значення відповідного елемента.
- $! x$  - надати відповідь на задачу - одне дійсне число ( $1 \leq x \leq 10^9$ ) - наближене значення середнього арифметичного довжин усіх шляхів в таблиці. Після цього запиту ваша програма має припинити свою роботу.

Максимальна дозволена кількість запитів першого типу - 1500.

### Приклад

standard input	standard output
1	? 1 1
1	? 1000 1000
1	? 1 1000
1	? 1000 1
	! 1999.000

## Задача G. Майже все по 1000

Назва вхідного файлу: `standard input`  
Назва вихідного файлу: `standard output`  
Ліміт часу: `3 seconds`  
Ліміт використання пам'яті: `256 megabytes`

У Поліни є ще одна таблиця  $1000 \times 1000$ , в кожній комірці якої записане дійсне число від 0 до 1. Вона називає *шляхом* у цій таблиці послідовність комірок, що починається у лівому верхньому куті таблиці, складається із пересувань донизу та праворуч та закінчується у правому нижньому куті. Сума чисел, що записані у комірках, через які проходить шлях, називаються його довжиною.

Поліну цікавить середнє арифметичне довжин усіх шляхів в таблиці. Звичайно, вона просить вас порахувати цю величину.

Але, якби Поліна сказала вам усі числа в таблиці, то було б занадто легко. Тому ви можете запитати в неї лише 115000 чисел з таблиці.

Якщо ваша відповідь  $X$ , а реальне значення середнього арифметичного  $Y$ , то відповідь буде вважатися правильною, якщо  $|X - Y| \leq 1$ .

### Протокол взаємодії

Інтерактивна взаємодія відбувається одразу.

Ви можете використовувати такі запити:

- $? i j$  - дізнатись елемент на перетині рядка  $i$  та стовпця  $j$  ( $1 \leq i \leq 1000$ ,  $1 \leq j \leq 1000$ ). У відповідь ви отримаєте одне дійсне число з точністю 4 знака після десяткової крапки - значення відповідного елемента.
- $! x$  - надати відповідь на задачу - одне дійсне число ( $0 \leq x \leq 2000$ ) - наближене значення середнього арифметичного довжин усіх шляхів в таблиці. Після цього запиту ваша програма має припинити свою роботу.

Максимальна дозволена кількість запитів першого типу - 115000.

### Приклад

standard input	standard output
0.0000	? 1 1
0.0000	? 1 1000
0.0000	? 1000 1
0.0000	? 1000 1000
	! 0.00000000



## Задача Н. Дерево

Назва вхідного файлу: `standard input`  
Назва вихідного файлу: `standard output`  
Ліміт часу: `0.5 seconds`  
Ліміт використання пам'яті: `256 megabytes`

Поліна десь знайшла повне  $k$ -ічне дерево з  $n$  вершин. Усі його вершини пронумеровані різними цілими числами від 1 до  $n$ , але ця нумерація вам не відома. Поліна просить вас знайти, який номер має корінь дерева.

Ви можете поставити Поліні не більше за  $60 \cdot n$  питань вигляду: "Чи проходить простий шлях з вершини з номером  $a$  до вершини з номером  $c$  через вершину з номером  $b$ ?"

Чи вистачить вам інформації, щоб визначити номер кореня?

### Протокол взаємодії

У першому рядку вхідних даних подаються два цілих числа -  $n$  ( $3 \leq n \leq 1500$ ) та  $k$  ( $2 \leq k < n$ ). Гарантується, що кількість вершин  $n$  відповідає деякому повному  $k$ -ічному дереву.

Після того, як ці дані були зчитані відбувається інтерактивна взаємодія. Ви можете використовувати такі запити:

- `? a b c` - дізнатись, чи проходить шлях з вершини з номером  $a$  ( $1 \leq a \leq n$ ) до вершини з номером  $c$  ( $1 \leq c \leq n$ ) через вершину з номером  $b$  ( $1 \leq b \leq n$ ). У відповідь ви отримаєте число 1, якщо так, і 0, якщо ні.
- `! r` - надати відповідь на задачу - ціле число, що відповідає номеру кореня  $r$  ( $1 \leq r \leq n$ ). Після цього запиту ваша програма має припинити свою роботу.

Максимальна дозволена кількість запитів першого типу -  $60 \cdot n$ .

### Приклад

<code>standard input</code>	<code>standard output</code>
<code>3 2</code>	<code>? 1 2 3</code>
<code>0</code>	<code>? 1 3 2</code>
<code>1</code>	<code>! 3</code>

### Зауваження

Повним  $k$ -ічним деревом називається дерево, в якому усі листи мають однакову глибину, та усі вершини, крім листів мають рівно  $k$  синів.

## Задача I. Цікаві картки

Назва вхідного файлу:	standard input
Назва вихідного файлу:	standard output
Ліміт часу:	0.25 seconds
Ліміт використання пам'яті:	256 megabytes

У Поліни є набір з  $N$  карток. На кожній картці записані два цілих числа - *ідентифікатор* картки  $n_i$  ( $1 \leq n_i \leq N$ ) та *значення* картки  $a_i$  ( $0 \leq a_i \leq 10^9$ ). Відомо, що для кожного числа  $k$  ( $1 \leq k \leq N$ ) існує рівно одна картка з ідентифікатором  $k$ . Також відомо, що  $a_i < a_{i+1}$  для всіх  $i$  від 1 до  $N - 1$ .

Поліна розклала картки на столі в один ряд за зростанням  $a_i$ . Тепер вона хоче для заданого  $X$  знайти ідентифікатор картки, яка має найменше значення серед тих карток, значення яких перевищують  $X$ , або дізнатись, що такої картки немає. Вона просить вас розв'язати цю задачу.

Але, звичайно, вона не хоче розповідати вам всю інформацію про картки. Ви можете декілька разів сказати їй ідентифікатор картки, а у відповідь вона скаже вам значення цієї картки та ідентифікатор картки, що лежить праворуч. Також вона сказала вам ідентифікатор найлівоїшої картки.

Чи зможете ви все ж таки відповісти на питання Поліни?

### Протокол взаємодії

У першому рядку вхідних даних подається три цілих числа  $N$  ( $1 \leq N \leq 50000$ ),  $X$  ( $0 \leq X \leq 10^9$ ) та  $n_1$  ( $1 \leq n_1 \leq N$ ) - кількість карток у наборі, задане ціле число з умови та ідентифікатор найлівоїшої картки відповідно.

Після того, як ці дані були зчитані відбувається інтерактивна взаємодія. Ви можете використовувати такі запити:

- ?  $n_i$  - дізнатись значення картки з ідентифікатором  $n_i$  ( $1 \leq n_i \leq N$ ) та ідентифікатор картки, що лежить праворуч. У відповідь ви отримаєте два цілих числа розділених пробілом - значення картки  $a_i$  ( $0 \leq a_i \leq 10^9$ ) та ідентифікатор картки праворуч  $n_{i+1}$ . Якщо картки праворуч немає, то замість її ідентифікатора буде виведено число  $-1$ .
- !  $n_i$  - надати відповідь на задачу - ідентифікатор картки  $n_i$  ( $1 \leq n_i \leq N$ ), що має найменше (серед усіх карток) значення  $a_i$ , яке перевищує  $X$ . Якщо такої картки не існує, замість  $n_i$  виведіть число  $-1$ . Після цього запиту ваша програма має припинити свою роботу.

Максимальна дозволена кількість запитів першого типу - 2000.

### Приклади

standard input	standard output
3 5	? 1
50 2	? 2
100 -1	? 3
1 1	! 1
1 0 1	? 1
0 -1	! -1

### Зауваження

В першому прикладі набір містить такі картки:

- $a_1 = 1, n_1 = 3$
- $a_2 = 50, n_2 = 1$
- $a_3 = 100, n_3 = 2$

Отже, картка з найменшим значенням, що більше за 5 є картка з ідентифікатором 1.

## Задача J. Встигни вгадати

Назва вхідного файлу: `standard input`  
Назва вихідного файлу: `standard output`  
Ліміт часу: `0.25 seconds`  
Ліміт використання пам'яті: `256 megabytes`

Поліна загадала деяке ціле число  $x$  ( $1 \leq x \leq N$ ), а ви маєте його відгадати. Для цього ви можете ставити їй питання вигляду "чи правда, що загадане число знаходиться на відрізку від  $l$  до  $r$  (обидва кінці включно)?".

Але не треба забувати, що Поліна - дівчина, і досить часто змінює свою думку, так само як і загадане число. Точніше, перед кожним вашим питанням вона змінює своє число (нехай воно було  $x$ ) на деяке число з відрізка  $[max(1, x - K); min(N, x + K)]$ .

Ваша задача полягає в тому, щоб вгадати поточне число Поліни, до того як вона знову його змінить.

### Протокол взаємодії

У першому рядку вхідних даних подаються два цілих числа  $N$  ( $1 \leq N \leq 10^{18}$ ) та  $K$  ( $1 \leq K \leq 10$ ).

Після того, як ці дані були зчитані відбувається інтерактивна взаємодія. Ви можете використовувати такі запити:

- $? l r$  - дізнатись чи знаходиться поточне число Поліни на відрізку від  $l$  до  $r$  ( $1 \leq l \leq r \leq N$ ). У відповідь ви отримаєте одне ціле число - 0, якщо ні, та 1, якщо так.
- $! x$  - надати відповідь на задачу - число  $x$  ( $1 \leq x \leq N$ ), що Поліна загадала останнім. Після цього запиту ваша програма має припинити свою роботу.

Максимальна дозволена кількість запитів першого типу - 1500.

Поліна змінює своє число **перед** кожним запитом **першого** типу.

### Приклад

standard input	standard output
10 2	? 3 5
1	? 3 3
0	? 3 4
1	? 5 5
1	! 5