

Задача А. Заборонена сума

Назва вхідного файлу:	standard input
Назва вихідного файлу:	standard output
Ліміт часу:	1.5 seconds
Ліміт використання пам'яті:	512 megabytes

Припустимо у вас є мультимножина натуральних чисел S . Забороненою сумою даної множини будемо називати мінімальне ціле додатне число, що не може бути отримане з суми підмножини чисел цієї множини.

Вам дано масив A з n натуральних чисел, а також m запитів, кожен з яких визначається парою цілих чисел l та r ($1 \leq l \leq r \leq n$). Вам потрібно знайти заборонену суму для мультимножини отриманого послідовністю чисел A_l, A_{l+1}, \dots, A_r .

Формат вхідних даних

Перший рядок містить одне ціле число n ($1 \leq n \leq 10^5$) — розмір масиву.

Другий рядок містить n цілих чисел A_1, A_2, \dots, A_n ($1 \leq A_i \leq 10^9$). Також відомо, що $A_1 + A_2 + \dots + A_n \leq 10^9$.

Третій рядок містить одне ціле число m ($1 \leq m \leq 10^5$) — кількість запитів.

Кожен з наступних m рядків містить пару цілих чисел l та r ($1 \leq l \leq r \leq n$) — новий запит.

Формат вихідних даних

Для кожного запиту виведіть в окремому рядку відповідь.

Приклад

standard input	standard output
5	2
1 2 4 9 10	4
5	8
1 1	8
1 2	8
1 3	
1 4	
1 5	

Задача В. Запити на xor

Назва вхідного файлу: `standard input`
 Назва вихідного файлу: `standard output`
 Ліміт часу: `2 seconds`
 Ліміт використання пам'яті: `512 megabytes`

Дано пустий масив, у якого нумерація починається з одиниці, та декілька типів запитів:

- Тип 0: Додати ціле число x в кінець масиву.
- Тип 1: На проміжку $l \dots r$ знайти число y таке, що значення $x \text{ xor } y$ максимальне.
- Тип 2: Видалити останні k чисел з масиву.
- Тип 3: На проміжку $l \dots r$ порахувати кількість чисел, менших або рівних x .
- Тип 4: На проміжку $l \dots r$ знайти k -е найменше число.

Формат вхідних даних

Перший рядок містить одне ціле число m ($1 \leq m \leq 500000$) — кількість запитів.
 Кожен з наступних m рядків містить запит в одному з наступних форматів:

- Запит типу 0 має вигляд $0 \ x$.
- Запит типу 1 має вигляд $1 \ l \ r \ x$.
- Запит типу 2 має вигляд $2 \ k$.
- Запит типу 3 має вигляд $3 \ l \ r \ x$.
- Запит типу 4 має вигляд $4 \ l \ r \ x$.

Для усіх запитів $1 \leq l \leq r \leq n$ та $1 \leq x \leq 500000$. Усі запити є коректними.

Формат вихідних даних

Для кожного запиту типу 1, 3 та 4 виведіть відповідь в окремому рядку.

Приклад

standard input	standard output
10	8
0 8	2
4 1 1 1	2
0 2	1
1 2 2 7	8
1 2 2 7	2
0 1	
3 2 2 2	
1 1 2 3	
3 1 3 5	
0 6	

Задача С. Соня та функція

Назва вхідного файлу: `standard input`
Назва вихідного файлу: `standard output`
Ліміт часу: 3 seconds
Ліміт використання пам'яті: 512 megabytes

У Соні є цілочисельний масив A довжини n та ціле число k .

Для кожної пари цілих чисел l, r таких, що $1 \leq l \leq r \leq n$, Соня визначила функцію $f(l, r) = \text{MIN}(l, r) \cdot \text{XOR}(l, r)$, де $\text{MIN}(l, r) = \min(A_l, A_{l+1}, \dots, A_r)$ та $\text{XOR}(l, r) = A_l \text{ xor } A_{l+1} \text{ xor } \dots \text{ xor } A_r$.

Соня хоче знайти k -е найменше значення функції $f(l, r)$. Це означає, що якщо Соня виписе всі значення функції f для усіх $n \times (n + 1)/2$ можливих пар значень, та відсортує їх по зростанню, то вона хоче знайти k -е значення серед них. Допоможіть Соні вирішити цю задачу.

Формат вхідних даних

Перший рядок містить пару цілих чисел n та k .

Другий рядок містить n цілих чисел — елементи масиву A .

$1 \leq n \leq 50000$

$1 \leq k \leq n * (n + 1)/2$

$1 \leq A_i \leq 50000$

Формат вихідних даних

Виведіть єдине число яке хоче знайти Соня.

Приклад

standard input	standard output
4 7 1 3 6 4	9

Задача D. Максимальний рядок

Назва вхідного файлу:	standard input
Назва вихідного файлу:	standard output
Ліміт часу:	2.5 seconds
Ліміт використання пам'яті:	512 megabytes

Розглянемо бінарний рядок S_0 довжини n , який спочатку заповнений нулями.

Необхідно виконати m операцій над цим рядком. Операція трансформує рядок S_{i-1} в рядок S_i . Таким чином, після k операцій рядок буде рівний S_k .

Операція задається парою цілих чисел (L_i, R_i) . Вона інвертує усі символи рядка в діапазоні $[L_i, R_i]$ (кінці відрізка включені в діапазон), тобто замінює усі символи 0 на 1, а усі символи 1 на 0.

Ваша задача — знайти серед усіх $m + 1$ рядків S_0, S_1, \dots, S_m , лексикографічно найбільший рядок, та вивести його.

Формат вхідних даних

Перший рядок містить пару цілих чисел n та m — довжина рядку та кількість операцій, відповідно.

Кожен з наступних m рядків містить пару цілих чисел (L_i, R_i) — опис операції.

$$1 \leq n, m \leq 100000$$

$$1 \leq L_i \leq R_i \leq n$$

Формат вихідних даних

Виведіть єдиний бінарний рядок — лексикографічно найбільший рядок серед усіх $m + 1$ рядків.

Приклад

standard input	standard output
10 10	1111100011
9 10	
6 10	
9 10	
1 8	
3 5	
3 3	
3 4	
3 9	
4 8	
7 7	

Задача E. DSU

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

Ваша задача — реалізувати **Persistent Disjoint-Set-Union**. Що це означає?

Про **Disjoint-Set-Union**:

Спочатку у вас є n елементів. Потрібно навчитись відповідати на 2 типи запитів:

+ **a b** — об'єднати множини, у яких лежать елементи a та b ;

? **a b** — сказати, чи лежать елементи a та b зараз в одній множині.

Про **Persistent**:

Тепер у нас буде декілька копій (версій) структури даних **Disjoint-Set-Union**.

Запити будуть виглядати так:

+ **i a b** — запит до i -ї структури, об'єднати множини, у яких лежать елементи a та b . При цьому i -а структура залишається незмінною, створюється нова версія, їй надається новий номер (**який? читайте далі**);

? **i a b** — запит до i -ї структури, сказати, чи лежать елементи a та b зараз в одній множині.

Формат вхідних даних

У першому рядку 2 числа N ($1 \leq N \leq 10^5$) та K ($0 \leq K \leq 10^5$) — кількість елементів та запитів. Спочатку усі елементи знаходяться в різних множинах. Ця початкова копія структури має номер 0.

Далі слідує K рядків, на кожному опис нового запиту. Формат запитів описаний в умові. Запити нумеруються числами від 1 до K .

При обробці j -го запиту нова версія отримує номер j .

Запитів до неіснуючих версій не буде.

Формат вихідних даних

Для кожного запиту виду ? **i a b** в окремому рядку треба вивести **YES** або **NO**.

Приклад

standard input	standard output
4 7	NO
+ 0 1 2	YES
? 0 1 2	YES
? 1 1 2	YES
+ 1 2 3	NO
? 4 3 1	
? 0 4 4	
? 4 1 4	

Задача F. Stack

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

Реалізуйте персистентний стек.

Формат вхідних даних

Перший рядок містить кількість подій n ($1 \leq n \leq 200000$). У рядку із номером $i + 1$ міститься опис події i :

t m — додати в кінець стеку із номером t ($0 \leq t < i$) число m ($0 \leq m \leq 1000$).

t 0 — видалити останній елемент стеку із номером t ($0 \leq t < i$). Гарантується, що стек t не порожній.

В результаті події i , описаної в рядку $i + 1$, утворюється стек із номером i . Спочатку існує лише порожній стек із номером 0.

Усі вхідні числа є цілими.

Формат вихідних даних

Для кожної операції видалення виведіть в окремому рядку видалений елемент.

Приклад

standard input	standard output
8	3
0 1	1
1 5	
2 4	
3 2	
4 3	
5 0	
6 6	
1 0	

Задача G. Оголошення на паркан

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

Соня нещодавно закінчила фарбування свого паркану. Паркан являє собою послідовність з n дощечок ширини 1 метр та довільної висоти. i -а дощечка має висоту h_i метрів. Сусідні дощечки слідуєть без проміжку.

Після фарбування Соня вирішила розмістити на паркані оголошення о його продажі. Оголошення буде вивпнено на листі бумаги прямокутної форми, і розміщено на паркані так, що бічні сторони листа паралельні дощечкам паркану, та краї листа збіжні з краями деяких дощечок. Соня визначила для себе наступні обмеження на оголошення:

1. Ширина оголошення після розміщення повинна бути рівно w метрів.
2. Оголошення повинно цілком знаходитись на ділянці паркану з l -ї по r -у дошку включно (в тому числі, неможна виходити за границі паркану по вертикалі).

Оголошення буде яскравим, тому Соня хоче, щоб висота оголошення була як можна більше.

Вам дано опис паркану та декілька запитів на розміщення оголошення. Для кожного запиту виведіть максимальну можливу висоту оголошення, яке можна розмістити на відповідній ділянці паркану при заданій фіксованій ширині оголошення.

Формат вхідних даних

Перший рядок містить одне ціле число n — кількість дощечок в паркані ($1 \leq n \leq 10^5$).

Другий рядок містить n цілих чисел h_i , розділених пробілом — висоти дошок ($1 \leq h_i \leq 10^9$).

Третій рядок містить одне ціле число m — кількість запитів ($1 \leq m \leq 10^5$).

В наступних m рядках слідує опис запитів, кожен з яких представлений трьома цілими числами l , r та w ($1 \leq l \leq r \leq n$, $1 \leq w \leq r - l + 1$) — відповідно границі ділянки паркану та ширина оголошення.

Формат вихідних даних

На кожний запит виведіть відповідь у окремому рядку — максимальна висота оголошення, яку можна отримати на відповідній ділянці паркану, при виконанні усіх умов.

Приклади

standard input	standard output
5	2
1 2 2 3 3	3
3	1
2 5 3	
2 5 2	
1 5 5	

Задача Н. Соня та книжки

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

Соня дуже полюбляє читати книжки. У неї вдома вже зібралось n книжок, впорядкованих певним чином на одній полиці. Кожна книжка має свій колір обкладинки та порядковий номер від 1 до n .

Соня вирішила зробити нові полиці для книжок, та розкласти їх так, щоб виконувались наступні умови:

- Кожна книжка розміщена рівно на одній полиці.
- Кількість різних кольорів книжок, що зустрічаються на кожній полиці окремо, не буде перевищувати певне число k .
- Усі книжки, що знаходяться на одній полиці, мають послідовні номери. Тобто, для будь-яких $1 \leq l \leq e \leq r \leq n$, якщо книжки із номерами l та r знаходяться на одній полиці, то і книжка e знаходиться на тій самій полиці.

Оскільки робити нові полиці не так вже й просто, Соня хоче щоб їх кількість була мінімальною, при певному параметрі k .

Соня зацікавилась питанням — скільки мінімум потрібно полиць, для кожного k від 1 до n . Допоможіть їй знайти відповідь.

Формат вхідних даних

Перший рядок вхідних даних містить єдине ціле число n ($1 \leq n \leq 10^5$) — кількість книжок у Соні.

Другий рядок містить послідовність з n цілих чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) — кольори книжок Соні, у порядку в якому вони розміщені на полиці.

Формат вихідних даних

В єдиному рядку виведіть n цілих чисел розділених пробілом. i -е число повинно бути рівним мінімальній кількості полиць, які треба зробити, якщо значення k буде рівним i .

Приклади

standard input	standard output
5 1 3 4 3 3	4 2 1 1 1
8 1 5 7 8 1 7 6 1	8 4 3 2 1 1 1 1

Задача I. Дужки

Назва вхідного файлу: `standard input`
 Назва вихідного файлу: `standard output`
 Ліміт часу: `3.5 seconds`
 Ліміт використання пам'яті: `512 megabytes`

Послідовність дужок називається правильною, якщо:

- Вона порожня.
- Вона складається з правильної дужкової послідовності, замкненої між парою відкриваючої та закриваючої скобок.
- Вона складається з двох правильних дужкових послідовностей, записаних одна за одною.

Наприклад, послідовності «`()()`» та «`((()))()`» є правильними дужковими послідовностями, а «`)(()`», «`(((((`» та «`()`» — ні.

Вам задано рядок, який складається з символів відкритої та закритої дужки. Ваше завдання — знайти кількість різних непорожніх підрядків цього рядка, які утворюють правильну дужкову послідовність. Іншими словами, вам треба порахувати кількість різних правильних дужкових послідовностей, які зустрічаються як підрядок даного рядку.

Формат вхідних даних

Перший рядок містить одне ціле число n ($1 \leq n \leq 500000$) — довжина рядка, який вам буде задано.

Другий рядок містить рядок довжини n , який складається з відкриваючих та закриваючих дужок.

Формат вихідних даних

Виведіть одне ціле число — кількість різних підрядків, які утворюють правильну дужкову послідовність.

Приклади

standard input	standard output
10 () () () ()	5
7) (() ()	3

Задача J. Соня та камінці

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

Соня любить подорожувати, тому вдома у неї зібралась колекція з n камінців з різних країн світу. Усі камінці розміщені в ряд та пронумеровані від 1 до n . Кожен камінець характеризується цілим числом a_i ($1 \leq a_i \leq 10^5$) — номер країни з якої його привезла Соня.

Соня хоче обрати деякий набір зі своїх камінців та подарувати його своїй подрузі Каті. Проте Соня знає, що цей набір не повинен містити більше ніж k камінців привезених з однієї країни, бо інакше набір не буде достатньо гарним.

У Соні є кілька варіантів для набору, кожен з яких задано парою цілих чисел (l_i, r_i) , що означає, що Соня збирається обрати лише камінці, що знаходяться на позиціях від l_i до r_i включно. Серед таких камінців Соня збирається обрати як можна більший набір, що не містить більше ніж k камінців, привезених з однієї країни.

Зверніть увагу, що параметри l_i та r_i задано в трохи зміненому вигляді. Детальніше в описі вхідних даних.

Формат вхідних даних

Перший рядок містить два цілих числа n та k ($1 \leq n, k \leq 10^5$).

Другий рядок містить послідовність з n цілих чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^5$) — номер країни з якої було привезено кожен камінь.

В третьому рядку записано єдине ціле число q ($1 \leq q \leq 10^5$) — кількість варіантів вибору набору.

Кожен з наступних q рядків містить пару цілих чисел x_i, y_i , що описують відповідний варіант набору.

Вам необхідно запам'ятовувати відповідь на останній оброблений варіант (назвемо це значення *last*). На самому початку $last = 0$. Числа l_i та r_i для i -го плану можна отримати за наступним алгоритмом:

- $l_i = ((x_i + last) \bmod n) + 1$;
- $r_i = ((y_i + last) \bmod n) + 1$;
- Якщо $l_i > r_i$, треба поміняти місцями значення l_i та r_i .

Формат вихідних даних

Виведіть q чисел. i -е число повинно бути рівним максимальному розміру *гарного* набору камінців при розгляданні i -го плану.

Приклад

standard input	standard output
6 2	2
1 1 1 2 2 2	4
5	1
1 6	3
4 3	2
1 1	
2 6	
2 6	