Задача A. Бінарна матриця

Дано матрицю a розміром $n \times n$, що складається з нулів і одиниць.

Будемо вважати, що рядки матриці пронумеровані зверху вниз від 1 до n, а стовпці зліва направо від 1 до n. Нехай $a_{i,j}$ — число на перетині i-го рядка та j-го стовпчика. Вам потрібно знайти кількість *гарних* підматриць цієї матриці.

Підматриця вважається *гарною*, якщо числа, записані в протилежних кутах, однакові.

В даній задачі підматрицею i_1, j_1, i_2, j_2 $(1 \leqslant i_1 < i_2 \leqslant n, 1 \leqslant j_1 < j_2 \leqslant n)$ будемо називати такі елементи $a_{i,j}$ заданої матриці, для яких виконуються умови $i_1 \leqslant i \leqslant i_2$ та $j_1 \leqslant j \leqslant j_2$. Зверніть увагу, що **довжина кожної сторони підматриці має бути принаймні** 2.

Формат вхідних даних

Перший рядок містить одне ціле число n $(1 \leqslant n \leqslant 1200)$ — розмір матриці.

Кожний з наступних n рядків містить по n символів «0» та «1».

Формат вихідних даних

Виведіть одне ціле число — кількість *гарних* підматриць матриці a.

Приклади

standard input	standard output
3 101 101 010	5
1 1	0
5 11001 10101 10111 00000 00010	22

Примітка

Координати протилежних кутів п'яти *гарних* підматриць у першому прикладі:

1. $(1, 1)$, $(2, 3)$.
2. $(1, 1)$, $(3, 2)$.
3. $(1, 2)$, $(3, 3)$.
4. $(2, 1)$, $(3, 2)$.
5. $(2, 2)$, $(3, 3)$.

Задача B. Граф

Дано граф з n вершинами та m ребрами. У i-й вершині графу записано число a_i. У кожній компоненті зв'язності вибирається одна головна вершина, після чого виконується операція побітового OR з усіма числами, записаними у відповідних обраних головних вершинах.

Необхідно мінімізувати результат побітового OR.

Формат вхідних даних

Перший рядок містить два цілих числа n та m $(1 \leqslant n, m \leqslant 2 \cdot 10^5)$ — кількість вершин та ребер.

Другий рядок містить n цілих чисел $a_1, a_2, \ldots, a_n$ $(0 \leqslant a_i < 2^{30})$ — числа, записані у вершинах.

Кожний з наступних m рядків містить по два цілих числа v_i та u_i $(1 \leqslant v_i, u_i \leqslant n)$ — вершини, між якими є ребро.

Не гарантується, що у графі немає кратних ребер та петель.

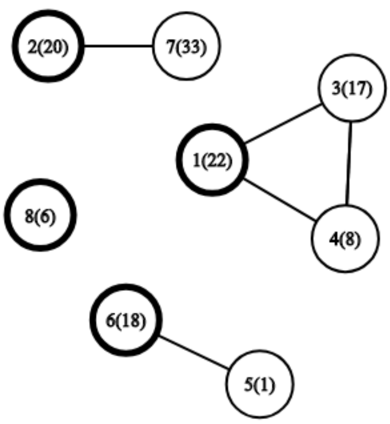
Формат вихідних даних

Потрібно вивести єдине число — мінімально можливе значення побітового OR.

Приклади

standard input	standard output
5 2 9 133 1 2 3 1 2 5 3	11
8 7 22 20 17 8 1 18 33 6 1 3 4 1 3 4 3 4 2 2 2 7 6 5	22

Примітка



У другому прикладі вигідно вибрати вершини з номерами 1, 2, 6 та 8.

Задача C. Два масиви

Дано дві послідовності цілих чисел a та b однакової довжини n.

Вартістю послідовностей a та b назвемо:

$$\sum_{i=1}^{n} a_i b_i$$

Іншими словами, $a_i b_i$ — числа a_i та b_i записані поспіль без пробілу. Наприклад, якщо $a_i = 562$, а $b_i = 12$, то $a_i b_i = 56212$.

Знайдіть мінімальну можливу *вартість* послідовностей a і b, якщо ви можете переставляти елементи послідовності a як вам заманеться.

Формат вхідних даних

Перший рядок містить одне ціле число n ($1 \leqslant n \leqslant 10^5$) — довжину послідовностей a і b.

Другий рядок містить n цілих чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^6$) — числа послідовності a.

Третій рядок містить n цілих чисел $b_1, b_2, \ldots, b_n$ ($1 \leqslant b_i \leqslant 10^6$) — числа послідовності b.

Формат вихідних даних

Виведіть одне ціле число — мінімальну можливу вартість двох послідовностей a та b, якщо можна переставляти елементи послідовності a.

Приклади

standard input	standard output
4 10 1 5 2 2 90 3 100	1545
8 123 51 245 50 2 59 2000 33 5 5 91 25 1 871 303 80	76061

Примітка

У першому прикладі однією з оптимальних перестановок послідовності a буде $[10, 2, 5, 1]$, тоді *вартістю* послідовностей буде $102 + 290 + 53 + 1100 = 1545$.

Задача D. Два числа

Вам дано два числа a та b.

Також у вас є масив простих чисел p розміром n.

За одну операцію ви можете взяти будь-яке число з масиву, помножити a або b на нього, та видалити це просте число з масиву. Вам треба відповісти, чи можна зробити так, щоб a дорівнювало b? А якщо можна, то потрібно знайти ще й мінімальну кількість операцій.

Зауважте, що необов'язково використовувати усі числа.

Формат вхідних даних

Перший рядок містить два цілих числа a та b ($2 \leqslant a, b \leqslant 10^8$).

Другий рядок містить одне ціле число n ($1 \leqslant n \leqslant 10^3$) — кількість елементів масиву.

Третій рядок містить n простих чисел $p_1, p_2, \ldots, p_n$ ($2 \leqslant p_i \leqslant 10^8$). Гарантується, що для кожного цілого i ($1 \leqslant i < n$) виконується $p_i \leqslant p_{i+1}$.

Формат вихідних даних

В першому рядку вам потрібно вивести «`YES`», якщо можна зробити a та b однаковими, інакше виведіть «`NO`». Якщо відповідь існує, в другому рядку потрібно вивести мінімальну кількість операцій.

Приклади

standard input	standard output
6 3 3 2 3 5	YES 1
6 3 2 3 5	NO

Примітка

У першому прикладі можна помножити друге число на 2, тоді $3 \cdot 2 = 6$.

У другому прикладі неможливо зробити числа рівними.

Задача E. Дерево і вірус

Вам дано дерево з n вершин. Кожна вершина має c_i тварин у ній. Коренем дерева є 1. Якщо заразити вірусом одну вершину у дереві, всі тварини, які живуть в ній стануть заражені. Також всім її синам передасться вірус. Зауважте, що предку вірус не передається. Якщо вершина заблокована, то вона не може бути зараженою та вона не може передавати вірус своїм синам.

Вам дано q запитів ($1 \leqslant q \leqslant 10^5$). Запити бувають двох типів:

1. Якщо дана вершина незаблокована — заблокувати її. Інакше розблокувати.
2. Заразити дану вершину та сказати, скільки тварин буде заражено. Кожен запит такого типу виконується незалежно один від одного, тобто на момент виконання цього запиту, ніяка вершина не є зараженою.

Формат вхідних даних

Перший рядок містить одне ціле число n ($1 \leqslant n \leqslant 10^5$) — кількість вершин у дереві.

Другий рядок містить n цілих чисел $c_1, c_2, \ldots, c_n$ ($1 \leqslant c_i \leqslant 10^9$) — кількість тварин у вершинах.

Кожний з наступних $n-1$ рядків містить по два цілих числа u і v ($1 \leqslant u, v \leqslant 10^5, u \neq v$) — вершини, між якими є ребро.

Наступний рядок містить одне ціле число q ($1 \leqslant q \leqslant 10^5$) — кількість запитів.

Кожний з наступних q рядків містить два цілих числа t та u ($1 \leqslant t \leqslant 2, 1 \leqslant u \leqslant n$) — тип запита та вершина запита.

Формат вихідних даних

Для кожного запиту другого типу ви маєте вивести кількість заражених тварин.

Приклад

standard input	standard output
7 1 2 3 4 5 6 7 1 2 1 3 2 4 2 5 3 6 3 7 7 1 3 2 1 1 5 2 2 1 3 2 1 2 3	12 6 23 16

Примітка

Вершини, які будуть заражені вірусом після другого дня $[1, 2, 4, 5]$, які містять 12 тварин. Після четвертого дня зараженими будуть вершини $[2, 4]$, які містять 6 тварин. Після шостого дня зараженими верщинами будуть $[1, 2, 3, 4, 6, 7]$, які містять 23 тварини. Після сьомого дня зараженими будуть $[3, 6, 7]$ вершини, які містять 16 тварин.

Задача F. K нульових бітів

Знайдіть будь-яке число, яке не менше за 10^n і у бінарному записі якого є відрізок довжиною **рівно** з k нулів, що з обох сторін **оточений одиницями**.

Наприклад, число $163_{10} = 10100011_2$ містить у собі відрізок із трьох нулів, оточений одиницями. Також це число містить у собі відрізок з одного нуля, але не містить з двох чи чотирьох нулів.

Формат вхідних даних

У першому рядку задано два цілих числа n та k ($1 \leqslant n \leqslant 1\,000$, $1 \leqslant k \leqslant 50$).

Формат вихідних даних

Виведіть будь-яке таке число у десятковому записі. Довжина відповіді не має перевищувати $2\,000$ символів. Можна довести, що відповідь завжди існує.

Приклади

standard input	standard output
2 3	163
3 5	2755593

Примітка

У другому прикладі: $2755593_{10} = 1010100000110000001001_2$. Це число більше за 10^3 та двійковому записі містить відрізки з нулів довжиною 1, 1, 5, 6, 2 символи. Через те, що серед них є і відрізок довжиною 5, то ця відповідь задовольняє умові.

Задача G. Кількість бітів

Дано число n. Знайдіть кількість одиниць у бінарному записі числа $2^n + n$.

Формат вхідних даних

Перший рядок містить одне ціле число t ($1 \leqslant t \leqslant 100$) — кількість тестів.

Кожний з наступних t рядків містить одне ціле число n ($0 \leqslant n \leqslant 10^{18}$).

Формат вихідних даних

Для кожного тесту вам потрібно вивести одне число в окремому рядку — кількість одиничних бітів числа $2^n + n$.

Приклад

standard input	standard output
3 3 1 2	3 2 2

Примітка

$2^3 + 3 = 11_{10} = 1011_2$ — три одиничних біта.

$2^1 + 1 = 3_{10} = 11_2$ — два одиничних біта.

$2^2 + 2 = 6_{10} = 110_2$ — два одиничних біта.

Задача H. Клас

Клас являє собою матрицю $n \times m$. Тобто всього $n \cdot m$ парт, за кожною з яких сидить рівно один студент.

Час змін! Усі вони хочуть змінити свої місця. Якщо студент сидить за партою (x, y), то він хоче пересісти на одну з парт: $(x+1, y)$, $(x, y+1)$, $(x-1, y)$, $(x, y-1)$. Якщо певної парти немає, то туди пересісти неможливо.

Вам потрібно визначити чи можуть усі студенти пересісти так, як вони хочуть.

Формат вхідних даних

Перший рядок містить одне ціле число t $(1 \leqslant t \leqslant 1\,000)$ — кількість тестів.

Кожний з наступних t рядків містить два цілих числа n та m $(1 \leqslant n, m \leqslant 10^5)$ — розміри класу.

Формат вихідних даних

Для кожного тесту вам потрібно вивести «TAK», якщо студенти можуть пересісти так, як вони хочуть, інакше виведіть «HI».

Зверніть увагу, що вам потрібно виводити букви латинського алфавіту, а не кирилицького.

Приклад

standard input	standard output
3 2 2 1 1 1 2	TAK HI TAK

Примітка

У першому прикладі студенти, які сидять в одному ряду, можуть помінятись місцями.

У другому прикладі всього один студент, який не можу нікуди пересісти.

У третьому прикладі два студенти можуть помінятись місцями.

Задача I. Круті підрядки

У вас є рядок s довжини n. Вам потрібно порахувати кількість пар (l, r) $(1 \leqslant l \leqslant r \leqslant n)$ таких, що підрядок $s[l \dots r]$ є `крутим`.

Підрядок називається `крутим`, якщо не існує такого символу, який зустрічається більше $\frac{l}{2}$ раз, де l — довжина підрядка.

Формат вхідних даних

У першому рядку задано одне ціле число n $(1 \leqslant n \leqslant 10^5)$ — кількість символів.

У другому рядку задано рядок з n літер латинського алфавіту в нижньому реєстрі від «`a`» до «`z`».

Формат вихідних даних

Виведіть кількість `крутих` підрядків.

Приклади

standard input	standard output
5 gccgc	4
6 abbcaa	10

Примітка

У першому прикладі `крутими` підрядками є $[1 \dots 2]$, $[1 \dots 4]$, $[3 \dots 4]$ та $[4 \dots 5]$.

У другому прикладі `крутими` підрядками є $[1 \dots 2]$, $[1 \dots 4]$, $[1 \dots 5]$, $[1 \dots 6]$, $[2 \dots 5]$, $[2 \dots 6]$, $[3 \dots 4]$, $[3 \dots 5]$, $[3 \dots 6]$ та $[4 \dots 5]$.

Задача J. Покриття

Дано масив a довжини n. Треба виконати m запитів: додати число x у мультимножину. Після кожного запиту потрібно сказати, скільки максимум елементів масиву a можна покрити елементами мультимножини. Кожен елемент мультимножини не можна використовувати більше одного разу. Елемент покриває інший елемент, якщо він більший або рівний за нього. Спочатку мультимножина порожня.

Формат вхідних даних

Перший рядок містить одне ціле число n ($1 \leqslant n \leqslant 200000$) — довжину масиву a.

Другий рядок містить n цілих чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^9$) — числа масиву a.

Третій рядок містить одне ціле число m ($1 \leqslant m \leqslant 200000$) — кількість запитів.

Кожний з наступних m рядків містить по одному цілому числу x ($1 \leqslant x \leqslant 10^9$) — запити.

Формат вихідних даних

На кожен запит виведіть в окремому рядку відповідь на цей запит.

Приклад

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

Примітка

Після першого та другого запиту можна покрити лише п'ятий елемент масиву a. Після третього запиту можна покрити третій та п'ятий елементи (а можна перший або другий замість третього). Після четвертого та п'ятого запиту можна покрити третій, четвертий та п'ятий елементи масиву a (можливі й інші покриття).

Задача K. Пончик

У Вас є масив із n чисел. Вам потрібно знайти кількість підмасивів цього масиву, для яких виконується правило: кожне число, що зустрічається хоча б раз на цьому підмасиві, має зустрічатись на ньому не менше ніж m разів. Підмасив — послідовність сусідніх позицій у масиві.

Формат вхідних даних

Перший рядок містить два цілих числа n та m ($1 \leqslant m \leqslant n \leqslant 5 \cdot 10^5$) — кількість чисел та мінімальна кількість разів, скільки мають зустрічатись числа.

Другий рядок містить n цілих чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 5 \cdot 10^5$) — числа масиву.

Формат вихідних даних

Виведіть одне число — відповідь на задачу.

Приклади

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

Примітка

У першому прикладі нам підійдуть такі підмасиви: $[2, 2]$, $[3, 3]$, $[1, 2, 2, 1]$, $[1, 2, 2, 1, 3, 3]$.

У другому прикладі нам підійдуть такі підмасиви: $[1, 2, 1, 1, 2, 2]$, $[1, 2, 1, 1, 2, 2, 1]$, $[1, 2, 1, 1, 2, 2, 1, 1]$, $[2, 1, 1, 2, 2, 1]$, $[2, 1, 1, 2, 2, 1, 1]$.

Задача L. Четвірки на відрізку

Дано послідовність цілих чисел a довжини n і m запитів двох типів:

1. «`! l r x`» — змінити a_i на x для кожного i на відрізку з l по r.
2. «`? l r`» — порахувати по модулю $10^9 + 7$

$$\sum_{l \leqslant i < j < k < w \leqslant r} a_i \oplus a_j \oplus a_k \oplus a_w$$

Вираз $x \oplus y$ означає застосування побітової операції `XOR` до чисел x і y. Дана операція існує у всіх сучасних мовах програмування, наприклад, в мовах *C++* і *Java* вона позначена як «`^`», а в *Pascal* як «`xor`».

Формат вхідних даних

Перший рядок містить ціле число n $(1 \leqslant n \leqslant 10^5)$ — довжину послідовності.

Другий рядок містить n цілих чисел $a_1, a_2, \ldots, a_n$ $(0 \leqslant a_i \leqslant 10^9)$ — числа послідовності.

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

Кожний з наступних m рядків описує один з двох запитів:

1. «`! l r x`» $(1 \leqslant l \leqslant r \leqslant n, 0 \leqslant x \leqslant 10^9)$.
2. «`? l r`» $(1 \leqslant l \leqslant r \leqslant n)$.

Гарантується, що буде принаймні один запит другого типу.

Формат вихідних даних

На кожен запит другого типу виведіть в окремому рядку відповідь на цей запит.

Приклади

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>4
1 9 4 5
6
? 1 3
? 1 4
! 1 1 9
? 1 4
! 2 4 5
? 1 4</pre></td><td><pre>0
9
1
12</pre></td></tr>
<tr><td><pre>10
335279264 849598327 822889311
446755913 526239859 548830120
181424399 715477619 342858071
625711486
10
! 5 5 647639160
? 4 7
! 2 6 545923679
? 2 8
! 7 9 389312924
? 5 10
! 6 10 561330066
? 1 8
! 3 10 786119025
? 2 7</pre></td><td><pre>375189142
259914062
838908849
461610148
401963966</pre></td></tr>
</table>

Примітка

Пояснення до першого прикладу:

Відповідь на перший запит 0, бо взагалі немає відповідних четвірок.

Відповідь на другий запит: $1 \oplus 9 \oplus 4 \oplus 5$ дорівнює 9.

Після третього запиту послідовність буде виглядати як $[9, 9, 4, 5]$.

Відповідь на четвертий запит: $9 \oplus 9 \oplus 4 \oplus 5$ дорівнює 1.

Після п'ятого запиту послідовність буде виглядати як $[9, 5, 5, 5]$.

Відповідь на шостий запит: $9 \oplus 5 \oplus 5 \oplus 5$ дорівнює 12.

Задача M. Розділіть послідовність

Вам дана послідовність з n невід'ємних цілих чисел. Вам треба розділити цю послідовність на $k+1$ непустих частин.

Спочатку, ви маєте одну частину. Щоб отримати $k+1$ частин, ви маєте повторити наступний алгоритм k разів:

1. Виберіть будь-яку частину, у якої більше одного елемента, та розділіть її на дві непусті частини довільного розміру.

2. До відповіді додайте добуток сум елементів двох нових частин.

Вам потрібно максимізувати фінальну відповідь.

Формат вхідних даних

Перший рядок містить два цілих числа n та k ($1 \leqslant n \leqslant 10^5, 1 \leqslant k \leqslant 200, k < n$) — довжину послідовності та кількість операцій розбиття.

Другий рядок містить n цілих чисел $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i \leqslant 10^4$) — елементи послідовності.

Формат вихідних даних

У першому рядку виведіть максимально можливий результат.

У другому рядку виведіть k чисел p_i ($1 \leqslant p_i \leqslant n-1$) — позиції, після яких треба розрізати послідовність. Тобто розріз проводився між p_i та p_{i+1}.

Приклади

standard input	standard output
7 3 4 1 3 4 0 2 3	108 1 3 4
10 2 1 2 3 4 5 6 7 8 9 10	999 6 8

Примітка

Пояснення першого прикладу:

У нас є початкова послідовність $[4, 1, 3, 4, 0, 2, 3]$.

Розрізаємо послідовність після першої позиції. Виходить дві нові послідовності $[4]$ та $[1, 3, 4, 0, 2, 3]$. Відповідь стає $4 \cdot 13 = 52$.

Далі розрізаємо послідовність після третьої позиції. Виходить $[4]$, $[1, 3]$, $[4, 0, 2, 3]$. Відповідь стає $52 + 4 \cdot 9 = 88$.

Розрізаємо після четвертої позиції. Виходить $[4]$, $[1, 3]$, $[4]$, $[0, 2, 3]$. Відповідь стає $88 + 4 \cdot 5 = 108$

Задача N. У пошуках медіани

Ван дано масив a довжиною n (n — непарне число) і q запитів до нього. Кожен запит задається двома числами: l та r. Ви маєте додати 1 до всіх елементів масиву на позиціях від l до r включно і після цього знайти медіану масиву та вивести її.

Медіана масиву з непарною довжиною — число, яке стояло б на $\frac{n+1}{2}$ позиції, якщо цей масив відсортувати.

Формат вхідних даних

Перший рядок містить два цілих числа n та q ($1 \leqslant n, q \leqslant 10^5$) — кількість чисел та запитів відповідно. Число n — непарне.

Другий рядок містить n цілих чисел $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i \leqslant 10^9$) — елементи масиву.

Кожний з наступних m рядків містить по два цілих числа l_i та r_i ($1 \leqslant l_i \leqslant r_i \leqslant n$).

Формат вихідних даних

Виведіть q рядків, де в i-му рядку має бути медіана масиву після i-того запиту.

Приклади

standard input	standard output
5 3 0 0 1 2 0 3 3 1 3 1 2	0 1 2
3 2 1 1 1 1 2 2 3	2 2

Примітка

Пояснення першого прикладу:

Позиція m, на якій знаходиться медіана у відсортованому масиві, рівна $\frac{5+1}{2} = 3$

Після першого запиту наш масив виглядає так $[0, 0, 2, 2, 0]$. Якщо його відсортувати, то він виглядає так $[0, 0, 0, 2, 2]$. На третій позиції у нас число 0, то й відповідь 0.

Після другого запиту масив виглядає наступним чином $[1, 1, 3, 2, 0]$. Якщо відсортувати, то це буде $[0, 1, 1, 2, 3]$. На третій позиції у нас 1.

Після третього запиту масив виглядає наступним чином $[2, 2, 3, 2, 0]$. Якщо відсортувати, то це буде $[0, 2, 2, 2, 3]$. На третій позиції у нас 2.

Задача O. Футбол

Микола Вікторович прийшов на футбольний матч. Як справжній футбольний фанат, він записував рахунок гри на листочку після кожного забитого голу. Наприклад, на листочку могли бути записані в такому порядку рахунки `0:1`, `1:1`, `2:1`, `3:1`, `3:2`.

Матч був досить цікавим, а голів було дуже багато, тому Миколі Вікторовичу набридло записувати всі рахунки гри. Але він ще є хорошим математиком, тому він запам'ятав суму всіх чисел в рахунках, які повинні були бути записані.

Вам Микола Вікторович, як своєму другу, сказав лиш цю суму, яку він запам'ятав, а от рахунки гри він забув. Та якщо ви просто скажете йому кількість забитих голів, його влаштує й це. Ваше завдання — знайти цю кількість голів, або повідомити, що Микола Вікторович не міг отримати таку суму і заплутався в розрахунках.

Формат вхідних даних

Перший рядок містить одне ціле число x ($1 \leqslant x \leqslant 10^9$) — суму, яку запам'ятав Микола Вікторович.

Формат вихідних даних

Якщо існує можлива кількість голів — виведіть цю кількість. В іншому випадку виведіть -1.

Приклади

standard input	standard output
6	3
1	1
5	-1

Примітка

В першому прикладі можливі такі рахунки: `0:1`, `0:2`, `1:2`. Забитих голів 3, сума чисел 6.

В другому прикладі можуть бути рахунки `0:1` або `1:0`.

У третьому прикладі неможливо отримати таке число.