

С [1]. Тайна Шай-Хулуда

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В далёких песчаных мирах, где дюны тянутся до самого горизонта, обитает древнее существо — Шай Хулуд. Это громадное, медленно движущееся существо, по виду напоминающее гигантского песчаного червя, чьи извивающиеся движения оставляют глубокие и длинные следы на поверхности барханов.

Исследователи, уставшие от бесплодных поисков, снарядили экспедицию: их задача — найти Шай Хулуда. Для этого они решили исследовать следы в песке, оставленные червем. Чтобы определить масштаб зоны его обитания нужно измерить максимальное расстояние между какими-то двумя следами. Это самое длинное расстояние, которое когда-либо отмечалось, и его знание может раскрыть тайну размеров и сил древнего существа.

Ваша задача — как раз помочь этим исследователям. Дана карта (список координат следов), и нужно определить: какая пара следов самая удалённая друг от друга.

Напомним, что расстояние между точками (x_1, y_1) , (x_2, y_2) может быть вычислено по формуле $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

Формат входных данных

В первой строке вводится целое число n ($2 \leq n \leq 100$) — число следов. В следующих n строках вводится n пар целых чисел x_i, y_i ($-1000 \leq x_i, y_i \leq 1000$) — координаты очередного найденного следа Шай-Хулуда.

Формат выходных данных

Выведите максимальное расстояние между двумя следами. Ваш ответ будет засчитан, если он отличается от верного не более, чем на 10^{-4} .

Примеры

стандартный ввод	стандартный вывод
2 51 52 52 51	1.4142135624
4 1 0 0 1 -1 0 0 -1	2.0000000000

[C] 2. Как много

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Дано два числа $S(1 \leq S \leq 100)$ и $X(1 \leq X \leq 100000)$.

Необходимо найти число троек неотрицательных чисел (a, b, c) , таких что $a + b + c \leq S$ и одновременно $abc \leq X$.

Формат входных данных

В первой строке записано одно целое число $T(1 \leq T \leq 10)$ — количество тестов.

В следующих T строках даны по два целых числа $S(1 \leq S \leq 100)$ и $X(1 \leq X \leq 100000)$.

Формат выходных данных

Выведите T строк — в i -й строке ответ на i -й тест.

Пример

стандартный ввод	стандартный вывод
4	20
3 10	56
5 20	10
2 5	35
4 15	

Замечание

Для $S = 3, X = 10$ существует 20 различных троек:

- $(0, 0, 0), (0, 0, 1), (0, 0, 2), (0, 0, 3)$ 4 — тройки
- $(0, 1, 0), (0, 1, 1), (0, 1, 2)$ — 3 тройки
- $(0, 2, 0), (0, 2, 1)$ — 2 тройки
- $(0, 3, 0)$ — 1 тройка
- $(1, 0, 0), (1, 0, 1), (1, 0, 2)$ — 3 тройки
- $(1, 1, 0), (1, 2, 0)$ — 2 тройки
- $(2, 0, 0), (2, 0, 1), (2, 1, 0)$ — 3 тройки
- $(3, 0, 0)$ — 1 тройка
- $(1, 1, 1)$ — 1 тройка

[C] 3. Бомбомбини Гузини

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Бомбомбини Гузини решил сменить имя, чтобы Бомбарди́ро Крокоди́ло было труднее его выследить. А именно, Бомбомбини решил заменить первую часть своего имени «Бомбом» на какое-нибудь натуральное число a .

При этом Гузини хочет, чтобы его имя не сильно поменялось, поэтому он хочет, чтобы a удовлетворяло следующим условиям:

1. a состоит из чётного числа цифр.
2. Первая половина в записи a совпадает со второй половиной в записи a .

Например, числа 33 и 147147 Гузини подойдут, а 333 и 147741 — нет.

Также по правилам страны, в которой живёт Гузини, в имени любого гражданина могут встречаться не любые натуральные числа, а только числа от 1 до N .

Ваша задача посчитать, сколько вариантов смены имени есть у Бомбомбини.

Формат входных данных

На вход подаётся единственное целое число N ($1 \leq N \leq 10^{12}$).

Формат выходных данных

Выведите одно целое число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
33	3
1390	13

Замечание

В первом примере входных данных Бомбомбини подойдут три числа: 11, 22, 33.

Во втором примере входных можно показать, что Бомбомбини подойдут 13 чисел. Например, одно из подходящих — 1313.

[C] 4. Великая стройка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Сережа и Саша решили все задачи, поэтому решили немного отвлечься от лекции. Сережа начал строить строки по определенному правилу:

- На нулевом шаге строка состоит из одной английской буквы **b**
- На шаге $3k + 1$, $k \geq 0$ он прибавляет слева от строки букву **a**, а справа букву **c**
- На шаге $3k + 2$, $k \geq 0$ он прибавляет наоборот слева от строки букву **c**, а справа букву **a**
- И на шаге $3k + 3$, $k \geq 0$ Сережа прибавляет справа и слева от строки букву **b**

Но иногда Сережа решает подшутить над Сашей — он может дать ему произвольную строку и сказать, что построил ее по правилу выше, хотя на самом деле это может быть не так. Саша хочет научиться проверять, обманывает его Сережа или нет.

Помогите Саше — по данной строке определите, могла ли она получиться в ходе построения, описанного выше, и если могла — то на каком шаге?

Формат входных данных

В первой строке входных данных дана длина строки N ($1 \leq N \leq 100$).

Во второй строке дана сама строка длины N , состоящая из маленьких латинских букв.

Формат выходных данных

Если строку можно получить в ходе построения, выведите номер шага, на котором она была получена. Если нельзя, то выведите -1 .

Примеры

стандартный ввод	стандартный вывод
5 cabca	2
4 abbc	-1

Замечание

В первом примере построение строки выглядит так:

- Шаг 0: «b»
- Шаг 1: «abc»
- Шаг 2: «cabca»

В итоге построение закончится на 2-м шаге.

Во втором примере построение строки «abbc» невозможно.

[C] 5. Кнут и пряник

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Денис ушел на каникулы и решил освежить в памяти собрание «Искусство программирования» Дональда Кнута. Он хочет прочитать подряд как можно больше томов, начиная с первого, потом второй, третий и так далее.

Всего на его полке N томов из «Искусства программирования», но случилось так, что не все тома идут подряд — среди их номеров могут быть пропуски. Денис взялся за дело серьезно и поэтому хочет читать тома именно подряд, без пропусков.

В случае нехватки какой-либо книги, он может пойти в местную библиотеку и поменять там два любых тома на один, нужный ему. Подобный обмен можно проводить сколько угодно раз, пока на руках у Дениса есть хотя бы две книги. Чтобы не терять время, Денис хочет поменять все необходимые книги до начала чтения первого тома.

Помогите Денису — определите, какое максимальное количество томов он сможет прочесть.

Формат входных данных

В первой строке дано число томов на книжной полке N ($1 \leq N \leq 3 \cdot 10^5$).

Во второй строке дан список из N томов по их номерам. Номера томов — целые числа от 1 до 10^9 , разделенные между собой пробелами.

Формат выходных данных

Выведите единственное число — какое максимальное количество томов Денис сможет прочесть.

Примеры

стандартный ввод	стандартный вывод
5 1 9 8 9 3	3
1 100	0

Замечание

В первом примере Денис может продать два тома 8 и 9 (или 9 и 9), обменяв их на том с номером 2. Тогда он сможет прочитать подряд 3 тома: 1, 2, 3.

Во втором примере у Дениса нет нужного числа томов на руках, чтобы обменять их в библиотеке на первый том, поэтому ответ 0.

[C] 6. Не все поймут

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Полина и Лиза любят посмеяться (а кто не любит?). Юмор у них тонкий, не все поймут. В частности, они любят следующую игру: Лиза пишет на доске пару целых неотрицательных чисел a и s , а Полина проверяет, существуют ли два целых неотрицательных числа x, y (возможно, совпадающих с a или s) таких, что

1. $x \& y = a$
2. $x + y = s$

Если такие два числа существуют, то Полина с Лизой считают пару смешной. И смеются. Иначе пара считается несмешной. И они не смеются.

Вы пришли в аудиторию и обнаружили, что на доске написано T пар чисел. Для каждой пары определите, посмеялись ли над ней Полина с Лизой, или нет.

$\&$ — это операция побитового «и». Она определяется так: пусть есть два целых неотрицательных числа a и b , записанных в двоичной системе счисления. Тогда в их побитовом «и» i -й разряд равен 1, если он равен 1 в a и в b , и 0 иначе. Например, если $a = 4_{10} = 100_2$ и $b = 6_{10} = 110_2$, то $a \& b = 100_2 = 4_{10}$.

Формат входных данных

В первой строке входных данных задано одно целое число T ($1 \leq T \leq 10^5$) — количество пар чисел на доске.

В последующих T строках записано по два целых числа через пробел — a и s в очередной паре. Все a и s удовлетворяют ограничениям: $0 \leq a, s < 2^{60}$.

Формат выходных данных

Выведите T строк. В i -й строке выведите «Yes», если Полина и Лиза посчитали i -ю пару смешной, и «No» иначе.

Пример

стандартный ввод	стандартный вывод
2	Yes
1 8	No
4 2	

Замечание

Для первой пары в примере из условия есть несколько x, y , удовлетворяющих условиям. Например, подойдёт $x = 3$ и $y = 5$.

Для второй пары можно показать, что не существует подходящих x, y .

[C-В'] 1. Зелье мудрости

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В одной далекой гильдии алхимиков юный подмастерье по имени Михаил хочет сварить особое зелье мудрости. Для этого ему нужно ровно s колб с редким концентратом.

В городе есть n лавок, где продаётся этот концентрат. В i -й лавке можно купить не более a_i колб, и каждая обойдётся Михаилу в c_i золотых монет.

Нужно узнать, за какую минимальную сумму золота Михаил сможет купить ровно s колб концентрата.

Известно, что в городе всегда можно купить нужное количество колб (т.е. их хватит во всех лавках вместе).

Формат входных данных

Первая строка содержит два целых числа n и s ($1 \leq n, s \leq 100\,000$) — количество лавок в городе и желаемое количество колб соответственно.

Следующие n строк содержат два целых числа c_i и a_i ($1 \leq c_i \leq 10^9$, $1 \leq a_i \leq 100\,000$), задающих цену и количество колб в i -й лавке.

Формат выходных данных

Выведите одно целое число — минимальную сумму золота, за которую можно купить ровно s колб концентрата.

Обратите внимание, что ответ может не помещаться в 32-битный тип данных, используйте 64-битные типы (например, `long long` в C++)

Примеры

стандартный ввод	стандартный вывод
2 10 4 6 2 6	28
4 10 5 10 1 3 2 1 4 3	32

Замечание

В первом примере необходимо купить 6 колб по 2 золотые монеты, а также 4 колбы по 4 монеты. Итоговая стоимость составит $6 \cdot 2 + 4 \cdot 4 = 28$ золотых монет.

[C-B'] 2. Собери команду мечты

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Как вы, вероятно, знаете игрок Solo недавно покинул Yandex Team в одной известной игре. Вы, как CEO Yandex team, хотите найти кого-нибудь ему на замену. Естественно, вы хотите выбрать самого сильного игрока.

Сейчас на рынке есть n игроков. Стоимость i -го игрока равна c_i . Конечно же, на профессиональной сцене нет игроков, которые хороши во всем одновременно. Поэтому для каждого игрока вы знаете, что он может исполнять l_i ролей. Для удобства обозначим каждую роль целым числом от 1 до m . Таким образом i -й игрок может исполнять роли $r_{i,1}, \dots, r_{i,l_i}$.

Скажем, что игрок i *хуже* игрока j , если выполнены следующие условия:

- $c_i \geq c_j$;
- Игрок j может играть на любой роли, на которой может играть i -й игрок;
- $c_i > c_j$ или игрок j может играть на роли, на которой не может играть i -й игрок.

Менее формально, игрок j должен уметь играть на всех ролях i -го игрока, но при этом быть дешевле или стоить столько же и знать еще какие-то роли.

Ваша задача определить, существует ли такая пара игроков, что один *хуже* другого.

Формат входных данных

Первая строка содержит целые числа n и m ($2 \leq n \leq 100$, $1 \leq m \leq 100$) — количество игроков и ролей соответственно.

Следующие n строк описывают игроков. i -я из них содержит целые числа c_i , l_i , $r_{i,1}, \dots, r_{i,l_i}$ ($1 \leq c_i \leq 100\,000$, $1 \leq l_i \leq m$, $1 \leq r_{i,1} < r_{i,2} < \dots < r_{i,l_i} \leq m$) — стоимость i -го игрока, количество ролей i -го игрока и сами роли, на которых может играть i -й игрок соответственно.

Формат выходных данных

Выведите «Yes», если существует такая пара игроков, что один *хуже* другого; иначе выведите «No» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
4 4 100 2 1 2 100 3 2 3 4 150 2 1 2 100 4 1 2 3 4	Yes
4 4 100 3 1 2 3 100 3 2 3 4 50 2 1 2 50 2 1 2	No

Замечание

В первом примере, например, третий игрок *хуже* первого, так как первый стоит меньше, но тоже умеет играть на позициях 1 и 2. Также первый игрок *хуже* четвертого, так как четвертый стоит столько же, но знает все роли первого, а также знает роли 3 и 4.

Во втором примере нет пары игроков такой, что один *хуже* другого. Третий и четвертый игроки дешевле первого и второго, но знают не все их роли. Первые два игрока стоят одинаково, но только первый знает роль 1, а также только второй знает роль 4. Третий и четвертый игроки стоят одинаково и каждый знает роли другого, но никто из них не знает каких-нибудь еще ролей.

[C-V'] 3. Зеркала

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На бескрайней каменной площади города выложена мостовая из бесконечной сетки плит. На одном участке мастера выделили прямоугольную арену из подряд идущих n рядов и m колонн. В каждую плитку этого прямоугольника вмонтировано дуиликое «лунное зеркало»: у него есть светлая и темная сторона. Сначала все зеркала повернуты светлой стороной вверх.

Над ареной проводят обряд. Для каждой плитки прямоугольника (ровно один раз для каждой) звучит удар колокола, и тогда «щелкают» зеркала в следующих местах: на самой целевой плитке и на всех восьми плитках, которые соприкасаются с ней ребром или углом (если такие плитки существуют в пределах площади). Каждое затронутое зеркало переворачивается на противоположную сторону.

Легко доказать, что конечный рисунок на площади не зависит от порядка, в котором ударяют по плитам. Требуется определить, сколько зеркал окажется темной стороной вверх после завершения обряда.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 10^9$) — размеры площади.

Формат выходных данных

Выведите одно целое число — количество зеркал, повернутых темной стороной вверх.

Примеры

стандартный ввод	стандартный вывод
1 1	1
1 6	4
4 5	6

Замечание

Разберем примеры из условия.

Первый тестовый пример.

Начальное положение зеркал: (C — светлая сторона)

C

После применения колокола к единственной клетке рисунок станет вида: (Т — темная сторона)

T

Второй тестовый пример. Начальное положение зеркал:

CCCCC

Применим колокол, например, только к клетке (1, 3):

CTTTCC

После применения колокола ко всем клеткам по одному разу рисунок станет вида:

CTTTTC

[C-B'] 4. Число подстрок

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Дана строка S ($|S| \leq 16$), состоящая из цифр, и два числа L, R ($0 \leq L \leq R \leq 10^{16}$). Пусть $f(x)$ — число вхождений числа x в строку S в качестве подстроки. Необходимо найти сумму $f(x)$ для всех $L \leq x \leq R$.

Формат входных данных

В первой строке задано число T ($1 \leq T \leq 1000$) — количество наборов входных данных.

Единственная строка каждого набора входных данных содержит строку S ($|S| \leq 16$), а также целые числа L и R ($0 \leq L \leq R \leq 10^{16}$). Гарантируется, что строка S состоит из цифр.

Формат выходных данных

Для каждого набора входных данных выведите сумму $f(x)$ для всех $L \leq x \leq R$.

Пример

стандартный ввод	стандартный вывод
5	1
5 1 10	5
123 1 100	4
100 0 10	1
999 100 1000	1
42 40 50	

Замечание

В первом наборе входных данных $f(5) = 1$, а в остальных точках функция равна нулю.

Во втором наборе входных данных $f(1) = 1, f(2) = 1, f(3) = 1, f(12) = 1, f(23) = 1$. В остальных точках функция равна нулю.

В третьем наборе входных данных $f(0) = 2, f(1) = 1, f(10) = 1$. В остальных точках функция равна нулю.

[C-B'] 5. Очередная игра из рекламы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вы, как обычно, тратите все свое свободное время за просмотром коротких вертикальных видео. К сожалению, на вашем любимом видеохостинге «ТРУБЫ» между видеороликами появляется реклама различных сомнительных игр. Одну из таких мы и рассмотрим!

У вас есть $2n$ кубиков. На каждом кубике написано целое число от 1 до n , при этом каждое число написано ровно на двух кубиках. Также у вас есть m полок. В начале раунда на i -й полке друг на друге лежат l_i кубиков. Вы знаете, что на j -м снизу из них написано число $c_{i,j}$. За одну операцию вы можете убрать два кубика, если выполняются следующие условия:

- Эти кубики лежат на разных полках;
- Эти кубики являются верхними на своих полках;
- На этих кубиках написано одно и то же число.

Цель игры — убрать все кубики. К сожалению, игру писали люди, которые не закончили даже параллель С, а потому вам необходимо определить, существует ли последовательность операций, позволяющая убрать все кубики.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n \leq 200\,000$, $2 \leq m \leq 200\,000$) — количество различных чисел и количество полок соответственно.

Следующие $2m$ строк содержат описание полок. Строки с номерами $2i$ и $2i+1$ содержат описание i -й полки. В $2i$ -й строке содержится положительное целое число l_i — количество кубиков на i -й полке. В $(2i+1)$ -й строке содержатся l_i целых чисел $c_{i,1}, \dots, c_{i,l_i}$ ($1 \leq c_{i,j} \leq n$) — числа, написанные на кубиках на i -й полке от самого нижнего к самому верхнему.

Гарантируется, что $\sum_{i=1}^m l_i = 2n$, а также что каждое число встречается ровно на двух кубиках.

Формат выходных данных

Если существует последовательность операций, позволяющая уничтожить все кубики, выведите «Yes»; иначе выведите «No» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
3 2 3 1 2 3 3 1 2 3	Yes
3 3 2 1 2 2 1 2 2 3 3	No
3 4 1 3 2 1 2 1 3 2 2 1	No

Замечание

В первом примере можно сначала убрать кубики с цифрой 3, потом кубики с цифрой 2 и в конце кубики с цифрой 1.

Во втором примере невозможно убрать все кубики, так как останутся два кубика с цифрой 3 на одной полке.

В третьем примере невозможно убрать все кубики, так как не получится убрать кубики с цифрами 1 и 2 (они мешают друг другу).

[C-B'] 6. Выходные на заводе

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Недавно Филипп вышел на завод с необычным графиком работы. Смена Филиппа на заводе будет длиться n дней, при этом каждый из дней, у которого ровно 9 делителей, считается выходным. Посчитайте, сколько выходных будет у Филиппа за одну смену на заводе.

Формат входных данных

В первой строке содержится одно целое число n ($1 \leq n \leq 4 \cdot 10^{12}$) — продолжительность смены на заводе.

Формат выходных данных

Выведите одно целое число — количество выходных за время одной смены.

Примеры

стандартный ввод	стандартный вывод
5	0
120	2
365	5

Замечание

В первом примере Филипп работает без выходных.

Во втором примере выходными считаются дни 36 и 100.

[В'-В] 1. Усиления

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В игре ДОКА-2 на параметры персонажа не влияют предметы в инвентаре, а все эффекты являются временными. В данной игре есть два вида эффектов, улучшающих параметры героя: *прибавки* (прибавляют фиксированное значение к базовой силе героя) и *проценты* (увеличивают силу героя на некоторый процент от базовой силы героя).

Пусть базовое значение силы героя равно b , и вы выбрали некоторые n эффектов прибавки с параметрами d_1, d_2, \dots, d_n и m эффектов процента с параметрами p_1, p_2, \dots, p_m . Тогда итоговое значение силы равно:

$$(b + d_1 + d_2 + \dots + d_n) \cdot \frac{100 + p_1 + p_2 + \dots + p_m}{100}$$

Обратите внимание, что каждый эффект можно применить не более одного раза.

Вам дано множество доступных эффектов прибавки и множество доступных эффектов процента. Ваша задача — выбрать некоторое количество эффектов, не превосходящее k , чтобы получить максимально возможную итоговую силу.

Формат входных данных

Первая строка содержит четыре целых числа b, k, x и y ($0 \leq b, k, x, y \leq 50\,000$) — базовое значение силы героя, максимальное количество эффектов, которые можно выбрать, количество доступных эффектов прибавки и количество доступных эффектов процента, соответственно.

Вторая строка содержит x целых чисел d_1, d_2, \dots, d_x ($0 \leq d_i \leq 50\,000$) — параметры эффектов прибавки.

Третья строка содержит y целых чисел p_1, p_2, \dots, p_y ($0 \leq p_i \leq 50\,000$) — параметры эффектов процента.

Формат выходных данных

В первой строке выведите два целых числа n и m ($0 \leq n \leq x, 0 \leq m \leq y, 0 \leq n + m \leq k$) — количество выбранных эффектов прибавки и эффектов процента, соответственно.

Во второй строке выведите n целых чисел i_1, i_2, \dots, i_n ($1 \leq i_t \leq x$) — номера выбранных эффектов прибавки. Все номера должны быть попарно различны.

В третьей строке выведите m целых чисел j_1, j_2, \dots, j_m ($1 \leq j_t \leq y$) — номера выбранных эффектов процента. Все номера должны быть попарно различны.

Эффекты прибавки пронумерованы целыми числами от 1 до x в порядке ввода, а эффекты процента пронумерованы целыми числами от 1 до y в порядке ввода.

Если существует несколько оптимальных решений, выведите любое из них.

Примеры

стандартный ввод	стандартный вывод
10 2 2 2 10 10 10 10	2 0 2 1
10 3 3 3 5 3 1 20 10 5	2 1 1 2 1

Замечание

Рассмотрим все варианты эффектов в первом примере:

- Если взять оба эффекта прибавки, то сила будет равна $(10 + 10 + 10) \cdot \frac{100}{100} = 30$

- Если взять один эффект прибавки и один эффект процента, то сила будет равна $(10 + 10) \cdot \frac{100+10}{100} = 22$
- Если взять оба эффекта процента, то сила будет равна $(10) \cdot \frac{100+10+10}{100} = 12$

Таким образом, для максимальной силы необходимо взять два эффекта прибавки.

[В'-В] 2. Великая уравниловка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

После Великой Левелистической революции в молодой стране с большими амбициями, ранее называвшейся Берляндией, полностью изменился уклад жизни. В жизни граждан появились две ключевые вещи: уравниловка и очереди. Великий Вождь революции, как водится, лучше всех знает, как устранить сложившееся неравенство. Сегодня Вождь решил, помимо прочих дел, обратить внимание на проблему распределения швейных машинок в молодой республике.

К счастью, ни комитеты равенства, ни народные коллегии восстановления справедливости вас не осудили, и вы смогли пробиться в советники к Вождю.

Итак, чтобы получить машинку, необходимо отстоять очередь. Кибернетика, как известно, наука продажная и империалистическая. Поэтому вместо компьютера поддерживать очередь будете вы.

Вам предстоит выполнять следующие операции:

1. В конец очереди встает человек, и, согласно требованиям законодательства, фиксируется количество денег, которое он имеет при себе.
2. В регистратуре магазина швейных машинок сообщают о прибытии новой машинки, и первый человек в очереди получает её. Когда его просят отдать все деньги, вы должны точно вспомнить, сколько денег у него было, чтобы они пошли на благо революции. Иначе гражданин может забыть, сколько денег у него в карманах, и случайно не отдать всё. Вы должны помочь ему избежать подобной ошибки!
3. Вождь телеграфирует вам приказ произвести уравниловку — люди в очереди сортируются в порядке убывания количества денег, начиная от головы очереди и заканчивая её хвостом. Это позволяет самым бедным получить швейную машинку раньше, а богатым — позже.

Вам нужно выполнять операции и отвечать на запросы второго типа, сообщая количество денег у человека, который получает машинку. Гарантируется, что в момент получения машинки в очереди есть хотя бы один человек.

Очередь сортируется только в моменты уравниловки. В остальное время она функционирует как обычная очередь. Если вас интересует, почему, представьте, что сделают ещё не привыкшие к идеям равенства граждане с мотыгами, когда решат, что уравниловка — это ваша прихоть, а не приказ Вождя.

Формат входных данных

В первой строке задано количество событий n ($1 \leq n \leq 2 \cdot 10^5$).

В следующих n строках описаны события, каждое из которых относится к одному из трёх типов:

1. **talon x** — человек с количеством денег x ($0 \leq x \leq 10^9$) встает в конец очереди.
2. **mashinka** — приходит новая машинка, и первый человек в очереди получает её, выходя из очереди.
3. **level** — производится уравниловка, при которой люди в очереди сортируются в порядке убывания количества денег.

Формат выходных данных

Для каждого события второго типа (**mashinka**) выведите количество денег у человека из головы очереди, который получает машинку.

Примеры

стандартный ввод	стандартный вывод
8 talon 100 talon 75 talon 70 talon 90 level mashinka talon 0 mashinka	70 75
9 talon 150 talon 150 talon 100 mashinka level mashinka talon 124 level mashinka	150 100 124

Замечание

Разберем случай в магазине из первого примера. Будем рассматривать состояние очереди после каждого события:

1. Приходит человек с 100 деньгами (Новыми Советскими бурлями). Очередь выглядит так: 100.
2. Приходит человек с 75 бурлями. Очередь выглядит так: 100, 75.
3. Приходит человек с 70 бурлями. Очередь выглядит так: 100, 75, 70.
4. Приходит человек с 90 бурлями. Очередь выглядит так: 100, 75, 70, 90.
5. Вождь вспоминает о швейных машинках и телеграфирует произвести уравниловку. Очередь выглядит так: 70, 75, 90, 100.
6. Приходит новая машинка, ее получает человек с 70 бурлями. Вы экспроприируете 70 бурлей у кулака. После этого очередь выглядит так: 75, 90, 100.
7. Приходит бедолага с 0 бурлями. Очередь выглядит так: 75, 90, 100, 0.
8. Приходит новая машинка, ее получает человек с 75 бурлями. В госбанк изымаются 75 бурлей.

[В'-В] 3. Слон и конь

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Ленивый слон по имени Г ни в какую не хочет делать задачу на отбор! Заставить его делать задачи под силу только великому коню по имени М.

Г очень умело ищет причины что-то не делать. В частности, сейчас помеха выглядит следующим образом: Г не может сконцентрироваться, пока не узнает, сколько в его жилье существует простых маршрутов из его комнаты.

Жилье Г можно представить как неориентированный граф с n вершинами и m ребрами. Комната Г находится в вершине 1. Дополнительно известно, что ни одна из вершин в таком графе не имеет степень больше 10 (в противном случае Г бы просто не справился ориентироваться в таком сложном жилье).

Определите, сколько существует простых маршрутов (т.е. маршрутов без посещения одной и той же вершины дважды) из вершины 1. Г еще не знает чисел больше миллиона, поэтому если число простых маршрутов составляет x , то выведите $\min(x, 10^6)$.

Формат входных данных

Первая строка содержит числа n ($1 \leq n \leq 2 \cdot 10^5$) и m ($0 \leq m \leq \min(2 \cdot 10^5, \frac{n(n-1)}{2})$).

Каждая из следующих m строк содержит ребро (u, v) ($1 \leq u, v \leq n$).

Гарантируется, что в графе нет петель и кратных ребер. Обратите внимание, что связность графа не гарантируется.

Формат выходных данных

Выведите одно число — $\min(x, 10^6)$, где x — количество простых маршрутов в жилье Г из комнаты 1.

Примеры

стандартный ввод	стандартный вывод
6 4 1 2 1 3 3 4 5 6	4
5 10 1 2 2 3 3 4 4 5 1 3 2 4 3 5 1 4 2 5 1 5	65

Замечание

В первом примере есть 4 маршрута: (1), (1, 2), (1, 3) и (1, 3, 4).

[В'-В] 4. Бронзовый век

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На модовом сервере Minecraft генерируется n уникальных рудных жил. Каждая жила встречается только один раз.

В i -й жиле находится c_i руды меди и t_i руды олова.

Вы собираетесь запустить бронзовый автопарк механизмов (шестерни, трубы, корпуса) и для старта нужно накопить как минимум a руды меди и как минимум b руды олова.

Вы можете раскопать любую жилу полностью, но не более одного раза. Тут мир не вайпается.

Нужно понять, можно ли собрать требуемые запасы и меди, и олова. Если можно — какое минимальное количество жил придётся раскопать, чтобы набрать не меньше a меди и b олова.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 300$) — количество рудных жил.

Вторая строка содержит два целых числа a и b ($1 \leq a, b \leq 300$) — требуемое количество руд меди и олова соответственно.

Следующие n строк содержат два целых числа c_i и t_i ($1 \leq c_i, t_i \leq 300$), задающих количество руд меди и олова в i -й жиле соответственно.

Формат выходных данных

Если существует набор жил, позволяющий набрать требуемое количество руд, выведите минимальное количество жил; иначе выведите -1 .

Примеры

стандартный ввод	стандартный вывод
3 2 3 1 1 2 1 1 1	3
2 100 1 3 1 5 1	-1
3 1 1 2 2 3 3 2 1	1

Замечание

В первом примере необходимо раскопать все 3 жилы, чтобы получить хотя бы 2 руды меди и 3 руды олова.

Во втором примере не получится набрать необходимое количество меди, раскопав все жилы.

[В'-В] 5. Просто, как АВС

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Вам дана строка s из n букв a , b и c .

Назовем подпоследовательность этой строки красивой, если она имеет вид $a \dots a b c \dots c$, причем количество букв a и c должно быть одинаковым (возможно, нулевым). Например, строки «abc», «aaabccss» и «b» являются хорошими, а строки «aabc», «cba», «ac» и «abca» хорошими не являются.

Требуется ответить на q запросов. В запросе даются два целых числа l и r . Требуется найти размер максимальной подпоследовательности подстроки s от l до r включительно, являющейся красивой.

Формат входных данных

Первая строка содержит два целых числа n и q ($1 \leq n, q \leq 200\,000$) — длина строки и количество запросов соответственно.

Вторая строка содержит строку s длины n , состоящую из букв a , b и c .

Следующие q строк содержат два целых числа l и r ($1 \leq l \leq r \leq n$), задающих запрос на подстроке s от l до r включительно.

Формат выходных данных

Для каждого запроса выведите одно целое число — максимальный размер красивой подпоследовательности заданной подстроки. Если красивых подпоследовательностей у данной подстроки нет, выведите 0.

Пример

стандартный ввод	стандартный вывод
7 4	5
acbabcc	0
1 7	1
1 2	3
1 3	
2 6	

Замечание

Первый запрос задает всю строку, ее максимальная красивая подпоследовательность это «aabcc», образованная символами с номерами 1, 4, 5, 6, 7.

Подстрока во втором запросе это «ac», у нее нет красивых подпоследовательностей.

Подстрока в третьем запросе это «acb», ее максимальная красивая подпоследовательность это «b».

Подстрока в последнем запросе это «cbabc», ее максимальная красивая подпоследовательность это «abc», образованная символами с номерами 4, 5 и 6.

[В'-В] 6. Оценка за курс

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Миша изучает курс по алгоритмам в Высшей Школе Битовых операций. Итоговая оценка за курс вычисляется следующим образом: в каждом из n домашних заданий студент решает какое-то количество задач, затем студент выбирает любые k различных домашних заданий и вычисляет побитовое И количества сданных задач из выбранного множества домашних заданий.

К концу семестра Миша решил повысить свою оценку. В каждом домашнем задании у него уже решено сколько-то задач, а также осталось время чтобы решить ещё t задач из любых домашек (можно дорешивать как одну домашку, так и несколько разных, возможно, он дорешает меньшее количество задач). К сожалению, Миша круто решает задачи по алгоритмам, но не может понять какие задачи ему стоит дорешать. Ваша задача определить его максимальную оценку за курс.

Напомним, что *побитовое И* неотрицательных целых чисел a и b равно неотрицательному целому числу, у которого в двоичном представлении на определенной позиции находится единица тогда и только тогда, когда в двоичных записях a и b на этой позиции находятся единицы. Например, $3_{10} \text{ AND } 5_{10} = 0011_2 \text{ AND } 0101_2 = 0001_2 = 1_{10}$.

Формат входных данных

В первой строке входных данных содержится три целых числа n , m , k ($1 \leq k \leq n \leq 2 \cdot 10^5$, $0 \leq m < 2^{30}$) — количество домашних заданий, количество задач, которые Миша успеет дорешать и количество домашних заданий, которые учитываются в вычислении итоговой оценки.

Во второй строке содержится n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i < 2^{30}$) — количество задач, которые решил Миша в каждом из домашних заданий.

Формат выходных данных

Выведите одно целое число — максимальную итоговую оценку за курс после того как Миша дорешает не больше m задач.

Примеры

стандартный ввод	стандартный вывод
5 2 3 6 6 15 7 4	7
3 2 3 4 4 4	4

Замечание

В первом примере Миша может дорешать по одной задаче из первого и второго домашнего задания и выбрать для подсчета итоговой оценки домашние задания с номерами 1, 2 и 4

Во втором примере Мише выгодно не дорешивать задачи.

[В-А'] 1. Петя и Депляндия

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Петя живет в своей любимой стране Депляндии. Депляндия — огромная страна, в которой есть n городов и m двусторонних дорог, i -я из дорог соединяет города x_i и y_i . Также, на i -й дороге написано число c_i . Может быть такое, что пара городов соединена более чем одной дорогой. Петя очень любит свою страну за ее связность, в Депляндии можно добраться от любого города до любого другого города по существующим дорогам.

К огромному сожалению Пети, в Депляндии грядет перестройка. Петя очень хочет, чтобы Депляндия осталась связной, поэтому он возглавит перестройку. Процесс перестройки состоит из двух этапов:

1. Инвестиции. Для поддержки страны, Петя должен инвестировать в каждый город целое число золотых монет от 1 до n (в каждый город может быть инвестировано свое число золотых монет).
2. Разрушение. В стране остаются только такие дороги, что число, написанное на дороге совпадает **ровно** с одним из количеств монет, инвестированных в города, между которыми проведена дорога.

Помогите Пете определить, может ли Депляндия остаться связной после перестройки. В случае, если она может остаться связной, скажите Пете сколько нужно инвестировать в каждый из городов. Обратите внимание, что Петя очень богатый, поэтому ему не важна суммарная инвестиция, его интересует только связность.

Формат входных данных

В первой строке находятся два натуральных числа n и m ($2 \leq n \leq 10^5$, $n - 1 \leq m \leq 2 \times 10^5$) — количество городов и дорог.

Следующие m строк содержат по три натуральных числа x_i, y_i, c_i ($1 \leq x_i, y_i, c_i \leq n$) — описание i -й дороги. Гарантируется, что $x_i \neq y_i$.

Формат выходных данных

Если Петя не может оставить Депляндию связной, выведите «No» (без кавычек), иначе выведите n строк, в i -й из которых должно находиться число монет, инвестированных в i -й город.

Пример

стандартный ввод	стандартный вывод
5 6	1
1 2 4	4
5 1 2	1
2 1 1	4
3 5 1	2
3 2 1	
1 4 4	

[В-А'] 2. Аккуратные распилы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На факультете геометрии одного мерзкого, флегматичного и тоскливого института хранятся иллюстрации выпуклых многоугольников, координаты вершин которых являются целыми числами. Администрация факультета считает иллюстрацию тем лучше, чем больше существует способов *аккуратно распилить* ее на две части.

Будем считать аккуратным распилом многоугольника отрезок, соединяющий две его вершины таким образом, что обе получившиеся части исходного многоугольника непусты (имеют положительную площадь), а также площади обеих частей являются целыми числами.

Студентам в качестве экзамена была дана задача — найти количество аккуратных распилов данного выпуклого многоугольника. Разумеется, студенты мерзкого института не справились с этой задачей. Теперь ее придется решить вам.

Формат входных данных

Первая строка содержит одно целое число n ($4 \leq n \leq 200\,000$) — количество вершин многоугольника.

Каждая из следующих n строк содержит два целых числа x_i и y_i ($-10^9 \leq x_i, y_i \leq 10^9$) — координаты i -й вершины многоугольника.

Гарантируется, что многоугольник является выпуклым, а вершины перечислены в порядке его обхода по или против часовой стрелки.

Формат выходных данных

Выведите одно целое число — количество аккуратных распилов данного многоугольника.

Примеры

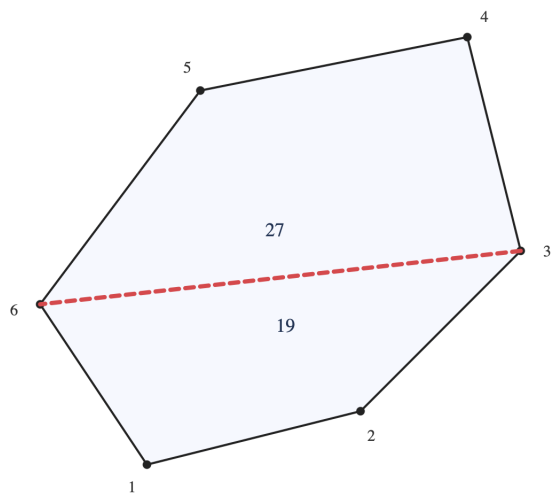
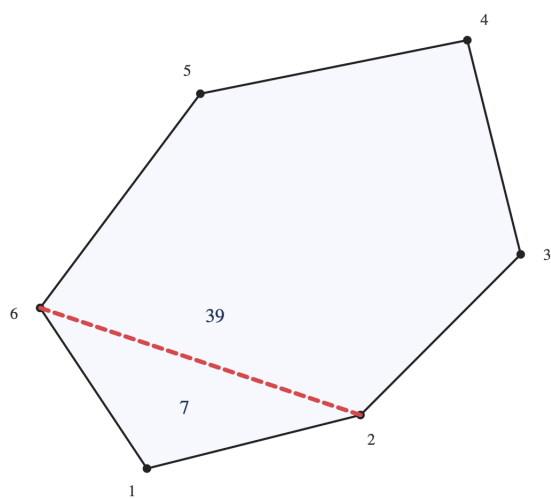
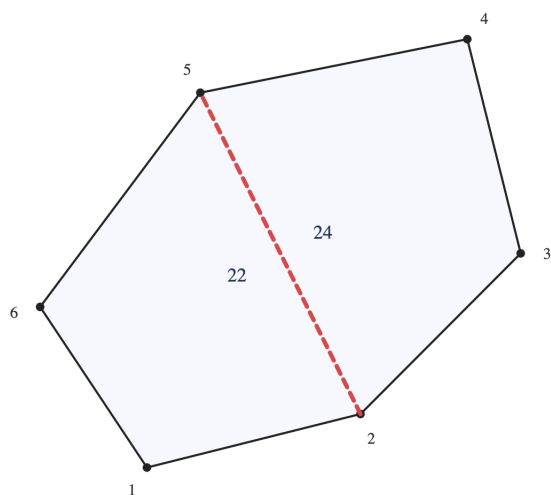
стандартный ввод	стандартный вывод
4 0 0 1 0 1 1 0 1	0
4 0 0 2 0 2 2 0 2	2
6 1 0 5 1 8 4 7 8 2 7 -1 3	3

Замечание

В первом примере существует два распила, однако они не являются аккуратными, так как делят многоугольник на части с площадью $\frac{1}{2}$.

Во втором примере оба распила являются аккуратными.

Иллюстрация к третьему примеру приведена ниже. На ней показаны аккуратные распилы, а также посчитаны площади полученных многоугольников.



[В-А'] 3. Граф-штурвал

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Петя обнаружил древний граф-штурвал с $n + 1$ вершинами, пронумерованными от 0 до n . Этот граф напоминает колесо, где центральная сокровищница (вершина 0) соединена с другими сокровищницами, образующими кольцо. Каждая вершина представляет сокровищницу, а рёбра — пути, за прохождение которых нужно заплатить золотые монеты.

Граф устроен следующим образом:

- Для каждого $i = 1, 2, \dots, n$ есть путь (неориентированное ребро) между центральной сокровищницей 0 и сокровищницей i , за который нужно заплатить p_i золотых монет.
- Для каждого $i = 1, 2, \dots, n$ есть путь между сокровищницей i и сокровищницей $i + 1$, за который нужно заплатить q_i золотых монет. (Здесь сокровищница $n + 1$ — это сокровищница 1)

Петя хочет разделить сокровищницы на две группы, чтобы можно было путешествовать только между сокровищницами разных групп (то есть сделать граф двудольным). Для этого он может закрыть некоторые пути, но за каждый закрытый путь с весом w он теряет w золотых монет. Помогите Пете определить, сколько золотых монет он должен потратить, чтобы граф-штурвал стал двудольным.

Формат входных данных

Первая строка содержит одно целое число n ($3 \leq n \leq 200\,000$) — количество вершин графа-штурвала минус один.

Вторая строка содержит n целых чисел p_1, p_2, \dots, p_n ($1 \leq p_i \leq 10^9$) — количество золотых монет за пути между сокровищницей 0 и сокровищницами $1, 2, \dots, n$.

Третья строка содержит n целых чисел q_1, q_2, \dots, q_n ($1 \leq q_i \leq 10^9$) — количество золотых монет за пути между сокровищницей i и сокровищницей $i + 1$ (где сокровищница $n + 1$ — это сокровищница 1).

Формат выходных данных

Выведите одно целое число — минимальную сумму золотых монет, которую Пете нужно потратить, чтобы граф-штурвал стал двудольным.

Примеры

стандартный ввод	стандартный вывод
4 1 1 1 1 5 5 5 5	2
5 1 2 3 4 5 3 3 2 6 1	7

[В-А'] 4. Игра на графе

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Дан неориентированный взвешенный связный граф из n вершин и m ребер.

Никита и Миша играют в игру. Сначала Никита красит каждую вершину графа в один из двух цветов (в том числе разрешается покрасить все вершины в один цвет). После этого Миша выбирает две различные вершины s и t , покрашенные Никитой в одинаковый цвет. Результатом игры считается расстояние между вершинами s и t в графе (минимальная сумма весов ребер на пути между s и t).

Никита стремится максимизировать итоговый результат, а Миша — минимизировать. Найдите итоговый результат игры, если Никита и Миша играют оптимально.

Формат входных данных

Первая строка содержит два целых числа n и m ($3 \leq n \leq 200\,000$, $n - 1 \leq m \leq \min(\frac{n(n-1)}{2}, 200\,000)$) — количество вершин и ребер графа соответственно.

Следующие m строк содержат три целых числа u , v и w ($1 \leq u < v \leq n$, $1 \leq w \leq 10^9$), задающих неориентированное ребро между вершинами u и v веса w .

Гарантируется, что граф является связным и все пары (u, v) попарно различны.

Формат выходных данных

Выведите итоговый результат игры при условии, что оба игрока играют оптимально.

Пример

стандартный ввод	стандартный вывод
3 3 1 2 3 2 3 4 1 3 8	7

Замечание

Пусть Никита покрасил вершины с номерами 1 и 3 в одинаковый цвет, а вершину с номером 2 — в другой цвет. Миша же, в свою очередь, может выбрать $s = 1$ и $t = 3$. Расстояние между этими вершинами равняется 7.

Можно показать, что Никита не может гарантировать себе больший ответ, поэтому результат игры равняется 7.

[В-А'] 5. Красивые числа

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Назовем положительное целое число **красивым**, если произведение его цифр делится на сумму его цифр. Найдите количество **красивых** чисел на отрезке $[L; R]$.

Формат входных данных

В единственной строке ввода находятся два целых положительных L и R , разделенные пробелом ($1 \leq L \leq R \leq 10^{12}$).

Формат выходных данных

Выведите единственное целое число — количество красивых чисел среди чисел от L до R .

Примеры

стандартный ввод	стандартный вывод
1 9	9
91 99	0
451 460	5

Замечание

В первом примере все числа на отрезке являются **красивыми**.

Во втором примере нет ни одного **красивого** числа на отрезке $[91; 99]$

В третьем примере есть 5 **красивых** чисел на отрезке $[451; 460]$ — 451, 453, 456, 459, 460.

[В-А'] 6. Русский дух против ящеров

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Известно, что древние русы боялись ящеров, которые периодически опустошали их земли и разрушали избы. Поэтому, когда началось строительство Великой избышки в Смоленске, люди стали всё больше беспокоиться, что ящеры разгневаются на дерзость человека и пошлют невиданный потоп, чтобы разрушить их труд.

Чтобы успокоить народ, местный поп обратился за помощью к Великому Оракулу Александру, который, выпив зелье богов, мог впадать в глубокий транс и разговаривать с ними. После многих дней и ночей переговоров и ожесточённых споров, которые можно было вести и улаживать только с помощью этого магического нектара, Оракул Александр вернулся к русам с планом строительства, одобренным Божественной комиссией по земным постройкам.

Избушка состоит из n горизонтальных слоёв деревянных блоков: первый слой содержит n блоков, второй — $n - 1$ блок, и так далее, вплоть до n -го слоя, который состоит из одного блока. Каждый блок имеет определённую прочность, выраженную целым положительным числом (определяется по множеству факторов: тип материала, плотность и т.д.).

Поскольку первый слой пирамиды **уже построен**, он остаётся без изменений. Каждый из следующих $n - 1$ слоёв бывает одного из двух типов: `min` или `max`, и блоки на нём укладываются так:

- Для слоя типа `min`: прочность каждого блока равна минимуму из прочностей двух блоков, на которых он стоит.
- Для слоя типа `max`: прочность каждого блока равна максимуму из прочностей двух блоков, на которых он стоит.

В знак благодарности русы решили установить Березку Оракулу Александру прямо на вершине пирамиды. Чтобы знать силу русского духа от березки, им нужно узнать прочность блока, находящегося на самом верхнем уровне.

Требуется по заданной структуре первого слоя и типу каждого из $n - 1$ последующих слоёв определить прочность блока на вершине избы.

Формат входных данных

В первой строке содержится единственное целое число n ($1 \leq n \leq 1\,000\,000$) — количество слоёв в избышке.

Во второй строке содержится n положительных целых чисел r_1, r_2, \dots, r_n ($1 \leq r_i \leq n$) — прочности блоков первого слоя.

В третьей строке содержится $n - 1$ символ из множества M, m — описание слоев, i -й символ описывает $i + 1$ слой

- `M` — слой типа `max`
- `m` — слой типа `min`

Формат выходных данных

Выведите одно целое число — прочность блока на n -м (верхнем) уровне.

Примеры

стандартный ввод	стандартный вывод
7 3 1 4 5 6 2 1 MmmMmM	5
2 2 1 m	1

[А'-А] 1. Урок математики

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Обычный день в школе Пети, он приходит на урок математики, на доске записано n целых чисел в столбик a_1, \dots, a_n , каждое из чисел имеет ровно m цифр, возможно с ведущими нулями. У чисел была своя особенность: они все расположены в строго возрастающем порядке, то есть $a_1 < a_2 < \dots < a_n$.

Но не все так просто, ведь это урок математики, поэтому учитель подходит к доске и стирает некоторые цифры у чисел и дает задание: «Скажите, сколькими способами можно восстановить стерты цифры так, чтобы полученные числа также шли в возрастающем порядке?».

За выполнение задания учитель пообещал 1337 упаковок мармеладных мишек, вишенки и другие невероятно вкусные сладости. Так как день обычный, то думать Пете сегодня не хочется, а сладостей хочется, поэтому он попросил вашей помощи, помогите решить задачу.

Так как ответ может быть большим, то Пете достаточно узнать остаток от деления числа способов на число $10^9 + 7$.

Формат входных данных

В первой строке содержатся два целых числа n, m ($1 \leq m \leq 40, 2 \leq n \leq 40$) — длина чисел и количество чисел.

В следующих n строках даются сами числа, в i -й строке записано число a_i , состоящее из цифр и, может быть, знаков ?, которые обозначают, что цифра в данном разряде стерта.

Формат выходных данных

В единственной строке выведите остаток от деления количества способов на число $10^9 + 7$

Примеры

стандартный ввод	стандартный вывод
3 2 ?3 ?? 18	18
2 1 1 1	0
5 5 ?1233 2???? 4???34 4?135 55555	13520020

[А'-А] 2. Дерево и НОД

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	8 секунд
Ограничение по памяти:	512 мегабайт

Дано неориентированное дерево, состоящее из n вершин, пронумерованных от 1 до n . Каждое из $n - 1$ рёбер соединяет пару различных вершин u_i и v_i ($1 \leq i \leq n - 1$).

Каждой вершине x приписано положительное целое число g_x .

Рассмотрим любые две различные вершины s и t . Пусть кратчайший путь между ними проходит через k вершин p_1, p_2, \dots, p_k , где $p_1 = s$, а $p_k = t$.

Определим функцию:

$$F(s, t) = k \cdot \gcd(g_{p_1}, g_{p_2}, \dots, g_{p_k}),$$

где $\gcd(a_1, a_2, \dots, a_m)$ — наибольший общий делитель этих чисел.

Требуется вычислить:

$$\sum_{1 \leq s < t \leq n} F(s, t)$$

и вывести остаток от деления результата на $10^9 + 7$.

Формат входных данных

Первая строка содержит одно целое число n ($2 \leq n \leq 10^5$) — количество вершин.

Вторая строка содержит n целых чисел g_1, g_2, \dots, g_n ($1 \leq g_i \leq 10^5$) — значения, написанные на вершинах.

Далее следуют $n - 1$ строк, каждая из которых содержит два целых числа u_i и v_i ($1 \leq u_i, v_i \leq n$) — описание i -го ребра. Гарантируется, что заданный граф является деревом.

Формат выходных данных

Выведите одно целое число — ответ на задачу по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
3 2 2 4 1 2 2 3	14
6 2 3 6 18 9 6 3 6 1 2 4 2 4 5 1 6	91
5 228 62 52 33 44 1 2 1 3 1 4 1 5	59

[А'-А] 3. Перестройка города

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Мэр города M потратил все деньги на гранты за олимпиады, поэтому теперь он хочет снести некоторые здания города и построить на их месте торговый центр, чтобы на этом заработать. Город M состоит из n домов, высота i -го дома равна h_i ($1 \leq i \leq n$). У мэра есть q планов по постройке торгового центра, i -й план задается отрезком домов от l_i до r_i включительно, т. е. для i -го плана нужно снести все дома от l_i до r_i . Однако сносить дома тоже не дешево. Чтобы снести дома на отрезке l до r , нужно сделать несколько операций следующего вида:

- Выбрать подотрезок от a до b ($l \leq a \leq b \leq r$) такой, что **равны высоты всех домов от a -го до b -го включительно**.
- Выбрать положительно целое число x .
- Вычесть из высот всех домов на подотрезке от a до b включительно число x . При этом высоты всех домов должны остаться неотрицательными, т. е. x должно быть не больше минимальной высоты дома на отрезке от a до b .

Стоимость сноса отрезка — это минимальное количество операций, которое нужно сделать, чтобы высоты всех домов на отрезке стали равны нулю. Мэр просит вас для каждого плана строительства торгового центра узнать стоимость сноса всех домов на отрезке от l_i до r_i .

Формат входных данных

В первой строке вводятся два целых числа n и q — количество домов в городе и количество планов постройки торгового центра ($1 \leq n, q \leq 2 \cdot 10^5$). Во второй строке вводятся n целых чисел h_1, h_2, \dots, h_n — высоты домов в городе ($1 \leq h_i \leq n$). В следующих q строках вводятся разделенные пробелом пары целых чисел l_i, r_i — планы постройки торгового центра ($1 \leq l_i \leq r_i \leq n$).

Формат выходных данных

Для каждого плана постройки ТЦ выведите в отдельной строке единственное число — минимальную стоимость сноса домов.

Пример

стандартный ввод	стандартный вывод
8 4	3
7 8 7 6 6 7 7 6	3
3 6	2
3 8	3
3 5	
1 4	

[А'-А] 4. Лабиринт с сокровищами

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В одном старом лабиринте, устроенном в виде дерева из n залов, спрятаны сокровища. Залы пронумерованы от 1 до n . Первый зал считается главным входом. Для каждого из залов (кроме входного) известен его родительский зал.

В каждом зале хранится по одной монете. В начале приключения в входном зале стоит фигурка-проводник. Приключение можно рассмотреть, как игру, в которой Дима и Сережа поочередно делают ходы. Первым ходит Дима. Игроки очень жадные, поэтому стремятся заполучить себе как можно больше монет. Более того, они очень умные, поэтому играют оптимально.

Правила игры следующие.

Если фигурка-проводник оказался в зале с монетой, то из-за своей жадности, игрок тут же заберет монету и его ход закончится. Если после этого, монеты в лабиринте закончились, то игра заканчивается.

Если же в зале нет монеты, то игроку предстоит перевести фишку-проводника в один из соседних залов. Если среди соседних залов есть зал с монетой, то игрок переведет фигурку в один из таких залов (он сам может решить в какой). Иначе не двигает фигурку в родительский зал. После того, как игрок подвинул фигурку-проводника, его ход заканчивается (то есть, если он попал в зал с монетой, то он не забирает монету).

Помогите Диме и определите, сколько монет он получит, если оба игрока играют оптимально.

Формат входных данных

В первой строке находится единственное натуральное число n ($2 \leq n \leq 10^5$) — количество залов.

Во второй строке находятся натуральные числа p_2, \dots, p_n ($1 \leq p_i < i$), где p_i — родительский зал для i -го зала.

Формат выходных данных

Выведите единственное число — сколько монет получит Дима при оптимальной игре обоих игроков.

Пример

стандартный ввод	стандартный вывод
7 1 1 2 1 4 5	4

[А'-А] 5. По кочкам

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

По ровной дорожке, по кочкам, по кочкам. По оврагам, по оврагам, по ухабам, по ухабам, на высокую горку, в ямку бух!

Вы проводите туры по горным тропам Гималаев. Гималаи представляют из себя последовательность из n стоящих в ряд гор. Исследования показали, что все высоты гор различны и являются целыми положительными числами от 1 до n , высота i -й горы равна p_i .

Недавно вам поступил запрос на необычный тур: клиент прислал вам строку s длины $n - 1$ из букв «U» и «D». Он хочет, чтобы вы нашли *подпоследовательность* a горных пиков максимальной длины (пусть она равна k), чтобы для любого $1 \leq j < k$ было выполнено $p_{a_j} < p_{a_{j+1}}$, если $s_j = \text{«U»}$ и $p_{a_j} > p_{a_{j+1}}$, если $s_j = \text{«D»}$.

Так как заказ еще не был оплачен, вы должны прислать клиенту число $(k - 1)$, то есть максимальную длину префикса запроса клиента, для которого можно найти подходящую подпоследовательность горных пиков.

Формат входных данных

В первой строке ввода находится единственное число n — количество горных пиков ($1 \leq n \leq 3 \cdot 10^5$).

Во второй строке находятся n чисел — массив p высот горных пиков (гарантируется, что p_i различны и $1 \leq p \leq n$).

В третьей строке ввода находится строка s из символов «U» и «D» длины $n - 1$ — запрос клиента.

Формат выходных данных

Выведите единственное число — ответ на задачу.

Пример

стандартный ввод	стандартный вывод
5 1 3 4 2 5 DUUD	2

Замечание

В примере можно найти подпоследовательность для префикса длины 2, то есть «DU» — для этого необходимо взять $a = [3, 4, 5]$.

[А'-А] 6. Шаи-Хулуд. Новые вызовы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

На планете Арракис обитают огромные песчаные черви, именуемые Шаи-Хулуд. Они могут перемещаться по планете, по городам и дорогам, их соединяющим. Дорожная инфраструктура планеты очень специфична. Города связаны ориентированными дорогами таким образом, что никакие две дороги не соединяют одну и ту же пару городов, никакая дорога не ведёт из города в себя же, а также для любой пары городов существует не более пятнадцати различных простых путей, соединяющих эти города.

Всего на Арракисе есть n городов и m дорог, при этом i -я дорога выходит из города с номером u_i и ведёт в город v_i и имеет длину c_i километров. Недавно на планете произошла песчаная буря, которая уничтожила все дороги планеты, а также k городов с номерами a_1, a_2, \dots, a_k . К сожалению, восстановить уничтоженные города уже не выйдет, поэтому Шаи-Хулуд решили хотя-бы восстановить и достроить дорожную сеть. Черви сторонники соблюдения традиций, духовных скреп и традиционных ценностей, а поэтому они решили, что в новой сети время движения должно остаться как раньше. А именно, для любой пары городов необходимо, чтобы время движения между ними по кратчайшему пути в новой сети было таким же, как время движения в старой дорожной сети.

Однако самым страшным последствием песчаной бури было не уничтожение дорог. Буря уничтожила главный дата центр планеты Арракис, где находились сервера языковой модели Червь GPT — уникального искусственного интеллекта, который опирался на фундаментальные традиционные ценности и защищал культурное пространство червей от внешнего вмешательства. Применение данной модели уже успело проникнуть во все сферы жизни червей, например даже на выборах черви голосовали в соответствии с рекомендациями государственного искусственного интеллекта. Из-за постоянной опоры на искусственный интеллект, черви перестали уделять много внимания интеллектуальному развитию, посвятив все свое время религии и патриотизму. После уничтожения искусственного интеллекта, червям придется самим восстанавливать дорожную систему планеты. Так как черви растеряли все свои когнитивные навыки, они решили обратиться к вам за помощью в восстановлении дорог планеты.

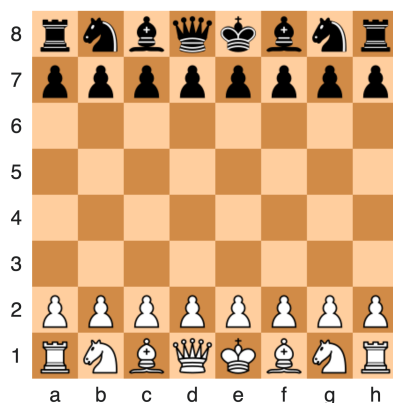
К сожалению, черви не понимают как можно отправить вам запрос о помощи (в прошлые годы запросы червей транслировались искусственным интеллектом). Поэтому черви приняли решение развивать свое образование и науку, чтобы суметь восстановить искусственный интеллект без помощи внешних специалистов. Начальным шагом для этого является игра в шахматы. Известно, что шахматы в игровой форме приносят огромную пользу, развивая когнитивные, социальные и эмоциональные навыки. Они улучшают логическое мышление, стратегическое планирование, память, концентрацию и способность к решению проблем.

Правила шахмат

Эта секция почти полностью повторяет правила шахмат с Википедии с небольшими отклонениями. Отклонения будут отдельно указаны в данных правилах.

В шахматы играют на доске, поделённой на равные квадратные клетки, или поля. Размер доски — 8×8 клеток. Вертикальные ряды полей (вертикали) обозначаются латинскими буквами от **a** до **h** слева направо, горизонтальные ряды (горизонтали) — цифрами от 1 до 8 снизу вверх; каждое поле обозначается сочетанием соответствующих буквы и цифры. Поля раскрашены в тёмный и светлый цвета (и называются, соответственно, чёрными и белыми) так, что соседние по вертикали и горизонтали поля раскрашены в разные цвета. Доска располагается так, чтобы ближнее угловое поле справа от игрока было белым (для белых это поле **h1**, для чёрных — поле **a8**).

У игроков в начале игры имеется по одинаковому набору фигур. Фигуры одного из игроков условно называются «белыми», другого — «чёрными». Белые фигуры окрашены в светлый цвет, чёрные — в тёмный. Сами игроки называются «белые» и «чёрные» по цвету своих фигур.



В каждый комплект фигур входят: король, ферзь, две ладьи, два слона, два коня и восемь пешек. В начальной позиции фигуры обеих сторон размещаются так, как показано на диаграмме выше. Белые занимают первую и вторую горизонтали, чёрные — седьмую и восьмую. Пешки расположены на второй и седьмой горизонталях соответственно.

Игра заключается в том, что игроки поочерёдно делают ходы. Первый ход делают белые. За исключением *особых ходов* (взятие на проходе и рокировка, описание данных ходов доступно ниже), ход заключается в том, что игрок перемещает одну из своих фигур на другое поле по следующим правилам:

- Фигуры (кроме коня) передвигаются по прямой линии, при этом все промежуточные поля между начальным и конечным должны быть свободны (на них не должно быть своих или чужих фигур). Исключением является ход коня.
- Ход на поле, занятое своей фигурой, невозможен.
- При ходе на поле, занятое чужой фигурой, она снимается с доски (такой ход называется взятием).
- Король ходит на соседнюю клетку по вертикали, горизонтали или диагонали.
- Ферзь ходит на любое расстояние по вертикали, горизонтали или диагонали.
- Ладья ходит на любое расстояние по вертикали или горизонтали.
- Слон ходит на любое расстояние по диагонали.
- Конь двигается на две клетки по вертикали и затем на одну клетку по горизонтали, или наоборот, на две клетки по горизонтали и на одну клетку по вертикали, тем самым движение коня напоминает заглавную букву «Г» кириллического или заглавную букву «L» латинского алфавитов. Другими словами, конь ходит на одно из полей, ближайших к тому, на котором он стоит, но не на той же самой горизонтали, вертикали или диагонали. При этом цвет поля, на которое ходит конь, всегда противоположен цвету поля, на котором он стоит. Конь отличается от остальных фигур также тем, что другие фигуры, стоящие на пути коня, не препятствуют его ходу (он «перепрыгивает» через них).
- Пешка может ходить только вперёд (направлением «вперёд» называется направление к восьмой горизонтали для белых или к первой для чёрных): без взятия — на одно поле вперёд по вертикали, а со взятием — по диагонали на одно поле вперёд-вправо или вперёд-влево. Если пешка находится на начальном поле (вторая горизонталь для белых и седьмая для чёрных), то она также может сделать ход без взятия на два поля вперёд. Когда пешка ходит на последнюю горизонталь (для белых — на восьмую, для чёрных — на первую), она заменяется по выбору игрока на любую другую фигуру того же цвета, кроме короля и пешки (то есть пешка может превратиться в ферзя, ладью, слона или коня). Такой ход называется *превращением пешки*. Превращение пешки является частью того хода, которым она перемещается на последнюю горизонталь. Таким образом, если, например, превращённая из пешки фигура угрожает королю

противника, то этот король немедленно оказывается под шахом. Нет никаких дополнительных ограничений на фигуру, в которую пешка может превратиться в результате превращения (то есть например на доске может стать больше одного ферзя).

Поле называется *битым*, если при своём ходе фигура могла бы взять находящуюся на этом поле фигуру противника (независимо от того, есть ли такая фигура на этом поле). Поле считается битым, даже если фактически ход фигурой туда невозможен, так как ставит под бой собственного короля.

Дополнительно существует два вида *особых ходов*:

- *Рокировка* — если король и одна из ладей того же цвета не двигались с начала игры, то король и эта ладья могут в один ход одновременно сменить положение (рокироваться). При рокировке король сдвигается на две клетки по направлению к ладье, а ладья ставится на поле между начальной и конечной позицией короля. Рокировка невозможна, если король или соответствующая ладья уже ходили. Рокировка временно невозможна, если между королём и ладьёй находится какая-либо фигура, а также если поле, на котором стоит король, или поле, которое он должен пересечь, или поле, которое он должен занять, находится под ударом одной из фигур противника. Обратите внимание, что при шахе король не может сделать рокировку, даже если это перенесет его на небитое поле. Также обратите внимание, что поле, на котором исходно стояла ладья или поле, через которое пройдет ладья но на которое не встанет король может быть битым.
- *Взятие на проходе* — когда пешка совершает ход на две клетки через битое поле, находящееся под ударом пешки противника, то ответным ходом она может быть взята этой пешкой противника. При этом пешка противника перемещается на битое поле, а взятая пешка снимается с доски (пример показан на диаграмме). Взятие на проходе возможно только пешкой непосредственно в ответ на ход пешки через битое поле, на следующих ходах оно уже не разрешено.

Король, находящийся на битом поле, называется «*стоящим под шахом*». Сделать ход, после которого король противника оказывается под шахом, значит «*объявить шах*». **Ходы, после которых король сделавшего ход остаётся или оказывается под шахом, запрещены.** Игрок, король которого находится под шахом, обязан немедленно его устранить.

Если король игрока находится под шахом и игрок не имеет ни одного хода, позволяющего устранить этот шах, этот игрок называется *получившим мат* и, соответственно, он терпит поражение. Цель игры и состоит в том, чтобы поставить мат королю противника.

Если игрок при своей очереди хода не имеет возможности сделать ни одного хода по правилам, но король игрока не находится под шахом, такая ситуация называется «*пат*».

Игра завершается выигрышем одной из сторон или ничьей. Ничья фиксируется в следующих случаях (**правила ничьи ниже незначительно отличаются от классических правил шахмат, прочтите их внимательно**):

- Поставлен *пат*.
- Троекратное повторение позиции (не обязательно в течение трёх ходов подряд), причём в понятие позиции в этом случае входит только расположение фигур (данное определение отличается от классических правил шахмат).
- *Правило 50 ходов* — в течение 100 последних полуходов (ходов одного из игроков) не было сделано ни одного взятия и ни одного хода пешкой.

Мат имеет приоритет перед ничьей, то есть если после хода игрока второй игрок находится в матовом положении (но не патовом), то игра завершается победой игрока, сделавшего ход, даже если какое-то из условий ничьи выполнено.

Итак, теперь вы знаете правила шахмат, а значит можно вернуться к Шаи-Хулуду. Шаи-Хулуд не собирается сыграть с вами партию в шахматы. Шаи-Хулуд нашли другую гораздо более увлекательную игру, которая очень популярна у людей — три в ряд! Однако Шаи-Хулуд решили добавить в игру азарта от покера, получив следующие правила игры:

Как и в традиционной игре три в ряд, в игре дано прямоугольное поле $n \times m$. Черви нормальные, поэтому n это число строк, а m — число столбцов. Соответственно, верхняя левая клетка поля имеет координаты $(1, 1)$, нижняя правая клетка имеет координаты (n, m) . В каждой клетке есть один шарик, покрашенный в цвет от 1 до k .

За время игры Шай-Хулуд выполняет q операций. Каждая операция заключается в обмене двух соседних (вертикально или горизонтально) шариков. После этого каждая непрерывная последовательность из не менее чем 3 шариков одного цвета в одной строке или столбце исчезнет. Назовем уничтожением такой набор из одного или нескольких исчезновений шариков.

После уничтожения все шарiki падают вниз под действием гравитации, создавая пустые клетки вверху поля (в данной вариации игры новые шарiki не падают сверху). Когда все шарiki упали, если остаются шарiki, которые можно уничтожить, происходит цепная реакция, продолжающая уничтожать их до тех пор, пока уничтожения невозможны. Одно уничтожение, за которым следует падение, называется «одним раундом уничтожения». Соответственно, за одну операцию может быть несколько раундов уничтожения.

Некоторые шарiki имеют особые свойства, которые вызывают специальные эффекты при их исчезновении. Всего существует 6 типов особых свойств:

1. Уничтожение убирает все шарiki в той же строке;
2. Уничтожение убирает все шарiki в том же столбце;
3. Уничтожение убирает все шарiki в той же строке и столбце;
4. Уничтожение убирает все шарiki в квадрате 3×3 с центром в этом шарике;
5. Уничтожение убирает все шарiki в квадрате 5×5 с центром в этом шарике;
6. Уничтожение убирает все шарiki того же цвета.

Срабатывание особого эффекта одного шарика может вызывать эффекты других шариков, но всё это происходит в рамках одного и того же раунда уничтожения, как цепная реакция, до падения шариков.

В игре каждая операция должна быть допустимой, то есть выбранные позиции должны быть соседними и непустыми, а также операция должна привести к уничтожению шариков. Если операция недопустима, она пропускается. Игра завершается после выполнения всех q операций.

Основной цвет допустимой операции определяется как цвет, который непосредственно уничтожается исходной перестановкой соседних шариков (не включая уничтожения от особых эффектов или падения). Очевидно, допустимая операция имеет 1 или 2 основных цвета.

В игре черви стремятся набрать как можно больше очков за свои операции. Обратите внимание, что при разработке системы очков Шай-Хулуд приняли большую дозу пряности с планеты Арракис, что имело свой эффект. В итоге система подсчёта очков была вдохновлена покером и состоит из 5 типов: очки за уничтожение + очки за цепочку + очки за комбинацию + очки за шаблон + очки за конец игры.

- **Очки за уничтожение:** Для каждой допустимой операции очки за i -й раунд уничтожения равны i , умноженному на сумму номеров цветов всех шариков, уничтоженных в этом раунде.
- **Очки за цепочку:** Если общее число раундов уничтожения для допустимой операции равно x , очки за цепочку равны $80 \cdot (x - 1)^2$.
- **Очки за комбинацию:** Для очков за комбинацию рассматриваются только уничтожения, вызванные не менее чем 3 последовательными шариками одного цвета в строке или столбце (игнорируя уничтожения от особых эффектов). Если уничтоженная связная область шариков одного цвета имеет размер x , очки за комбинацию равны $50(x - 3)^2$. Примеры: 4 шарика одного цвета в ряд дают 50 очков; 5 шариков одного цвета, образующих линию, крест или Т-образную форму, дают 200 очков; прямоугольник 2×3 из шариков одного цвета (который может появиться после падения) даёт 450 очков.

- **Очки за шаблон:** Каждые 5 допустимых операций рассчитываются очки за шаблон на основе основных цветов предыдущих 5 допустимых операций по правилам, вдохновленным покером (если операция имеет несколько основных цветов, берётся тот, который даёт наибольший балл по следующим правилам):
 - Старшая карта: все 5 цветов разные, очки равны $50 + \text{наибольший номер цвета среди всех карт}$;
 - Пара: 2 шарика одного цвета + 3 шарика разных цветов, очки равны $100 + \text{номер цвета пары} \times 2$;
 - Две пары: 2 пары шариков одного цвета + 1 другой цвет, очки равны $200 + \text{большой номер цвета пары} \times 2 + \text{меньший номер цвета пары}$;
 - Сет: 3 шарика одного цвета + 2 разных цвета, очки равны $300 + \text{номер цвета тройки} \times 3$;
 - Фул-хаус: 3 шарика одного цвета + 2 шарика другого цвета, очки равны $500 + \text{номер цвета тройки} \times 3 + \text{номер цвета пары}$;
 - Каре: 4 шарика одного цвета + 1 другой цвет, очки равны $750 + \text{номер цвета каре} \times 5$;
 - Пятёрка: все 5 шариков одного цвета, очки равны $1000 + \text{номер цвета пятёрки} \times 10$.
- **Очки за конец игры:** Если все q операций допустимы, в конце добавляется бонус 1000 очков. Если доска полностью очищена в конце игры, добавляется бонус 10 000 очков.

Правила подсчета очков оказались слишком сложными для Шай-Хулуда, и без своего искусственного интеллекта они не могут посчитать очки ни в одной из игр. Пришло время заканчивать условие этой задачи, и наверное вы уже догадались что вам надо сделать! Вам дан набор действий, совершенный Шай-Хулудом. Посчитайте итоговое количество очков, которые он заработает.

Формат входных данных

Первая строка входных данных содержит четыре целых числа n , m , k и q ($2 \leq n, m \leq 50$; $m + n > 4$; $2 \leq k \leq 100$; $1 \leq q \leq 1000$).

Далее следуют n строк, каждая из которых содержит m целых чисел $a_{i,j}$ ($1 \leq a_{i,j} \leq k$) — изначальные цвета шариков. Строки даны в порядке сверху вниз.

Далее следуют n строк, каждая из которых содержит m целых чисел $b_{i,j}$ ($0 \leq b_{i,j} \leq 6$) — начальное состояние особых эффектов шариков. $b_{i,j} = 0$ означает отсутствие особого эффекта.

Каждая из следующих q строк содержит четыре целых числа: $x_{i,1}$, $y_{i,1}$, $x_{i,2}$ и $y_{i,2}$ ($1 \leq x_{i,1}, x_{i,2} \leq n$; $1 \leq y_{i,1}, y_{i,2} \leq m$), обозначающих обмен шариков в координатах $(x_{i,1}, y_{i,1})$ и $(x_{i,2}, y_{i,2})$.

Гарантируется, что начальное состояние не содержит ситуаций прямого уничтожения.

Формат выходных данных

В единственной строке выведите одно число — итоговый балл, заработанный в игре.

Примеры

стандартный ввод	стандартный вывод
8 8 5 5 3 5 4 5 1 5 1 1 2 3 4 5 2 2 1 2 4 5 1 2 4 1 4 1 4 4 1 2 5 5 1 2 2 2 4 3 2 5 3 2 5 4 2 3 3 4 2 4 3 4 2 5 3 4 3 1 2 1 1 2 5 2 4 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 5 0 1 3 0 0 6 0 0 0 0 0 0 3 0 0 0 0 4 1 0 0 0 0 0 0 3 7 4 7 5 5 5 4 7 7 7 6 8 4 8 3 6 2 6 1	11692
8 8 5 8 3 5 4 5 1 5 1 1 2 3 4 5 2 2 1 2 4 5 1 2 4 1 4 1 4 4 1 2 5 5 1 2 2 2 4 3 2 5 3 2 5 4 2 3 3 4 2 4 3 4 2 5 3 4 3 1 2 1 1 2 5 2 4 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 5 0 1 3 0 0 6 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 1 8 2 7 3 7 4 7 3 7 3 6 4 7 4 6 5 5 5 4 7 7 7 6 8 4 8 3 6 2 6 1	684

Замечание

Суммы первых трёх типов очков после каждой операции составляют: 315, 417, 429, 435, 482. После 5-й операции рассчитываются очки за шаблон, при этом оптимальный шаблон — (1 2 4 2 4), дающий $200 + 4 \cdot 2 + 2 \cdot 1 = 210$ очков. В конце игры получены оба вида бонусов за окончание, что в сумме даёт общий счёт 11 692.

[А] 1. Линия огня

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Прошло несколько лет после ядерной войны. Когда-то шумный мегаполис превратился в руины, разделённые на кварталы. Большая часть зданий обрушилась, но местами уцелели многоэтажки — мрачные памятники прошлого, перегораживающие и путь, и обзор.

Город моделируется прямоугольной сеткой из r строк и c столбцов квадратных кварталов. Некоторые кварталы заняты уцелевшими зданиями; далее будем называть их застроенными. Сквозь застроенные кварталы нельзя пройти и в них нельзя заглянуть.

Вы — снайпер из небольшой группы выживших. Ваш наблюдательный пункт расположен в квартале $(1, 1)$ — там сохранились руины дома, дающие укрытие (эти руины не считаются застроенным препятствием). Вдали, в квартале (r, c) , патрулирует вражеский дозорный. Он обходит весь квартал, и, если вы готовы ждать, цель обязательно окажется в любой точке своего сектора.

Стрелять можно только с края вашего укрытия: начальная точка выстрела должна лежать на границе квартала $(1, 1)$. Линия выстрела не должна пересекать застроенные кварталы и не должна иметь с ними общих точек — даже касание границы запрещено. Формально: каждый квартал (i, j) — это замкнутый единичный квадрат $[i - 1, i] \times [j - 1, j]$; выстрел — отрезок между некоторой точкой на границе квартала $(1, 1)$ и некоторой точкой внутри или на границе квартала (r, c) . Отрезок не должен иметь общих точек ни с одним застроенным кварталом.

Определите, существует ли хотя бы одна траектория выстрела, соответствующая этим условиям.

Формат входных данных

Первая строка содержит три целых числа r , c и k — число строк, столбцов и застроенных кварталов ($2 \leq r, c \leq 1000$; $1 \leq k \leq 10^5$).

В следующих k строках даны пары r_i, c_i — координаты застроенных кварталов ($1 \leq r_i \leq r$, $1 \leq c_i \leq c$).

Гарантируется, что кварталы $(1, 1)$ и (r, c) не застроены и все координаты попарно различны.

Формат выходных данных

Выведите 1, если выстрел возможен, и 0 в противном случае.

Примеры

стандартный ввод	стандартный вывод
3 3 2 1 3 3 1	1
2 2 2 1 2 2 1	0
5 6 2 3 2 3 4	0
5 6 1 2 2	1

Замечание

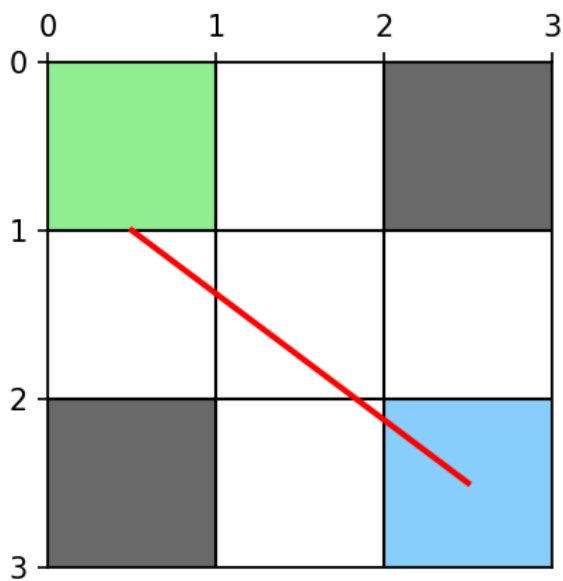


Рис. 1. Иллюстрация к примеру 1.

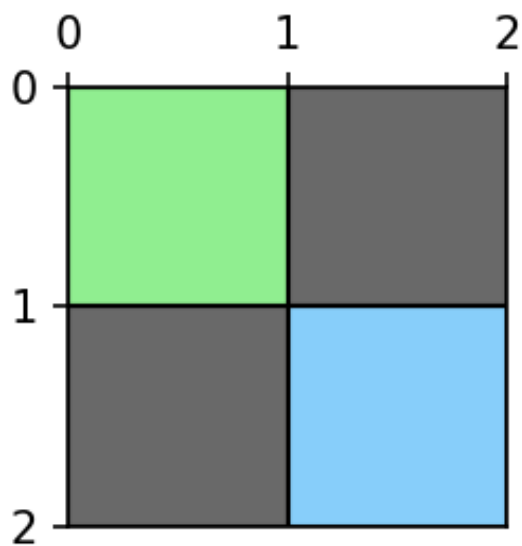


Рис. 2. Иллюстрация к примеру 2.

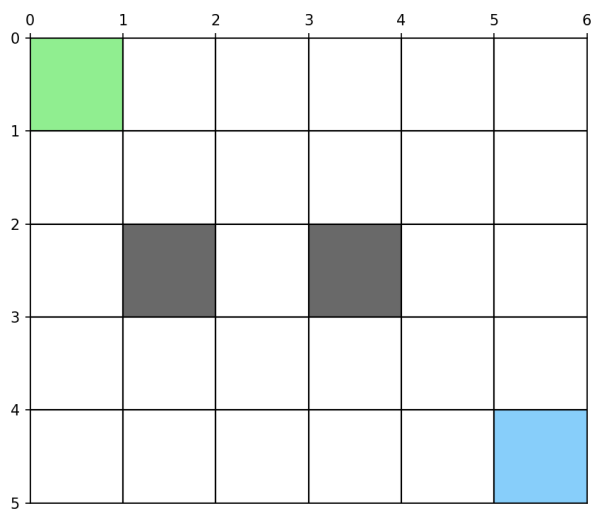


Рис. 3. Иллюстрация к примеру 3.

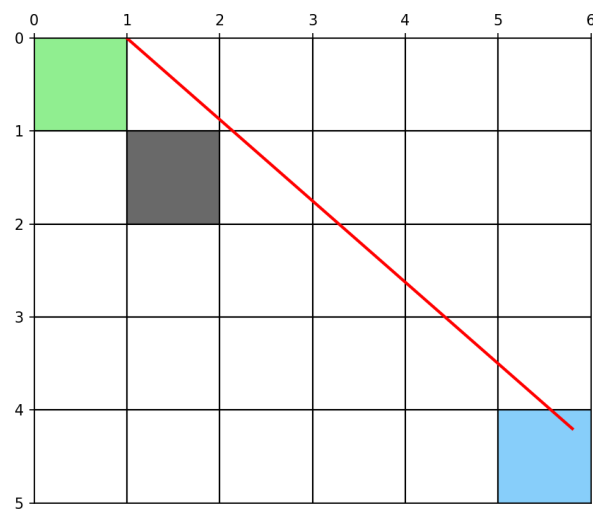


Рис. 4. Иллюстрация к примеру 4.

[A] 2. Камни SUPER HARD

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Гора состоит из площадок, соединенных между собой узкими проходами. На каждой площадке лежит какое-то количество камней. У разных площадок разный рейтинг и очевидно, что на площадках с более высоким рейтингом лежит большее количество камней.

В один солнечный день главный камень одной из площадок обратил внимание, что вокруг стало очень тесно. Но камни не хотят перемещаться на другие площадки, ведь от этого они скатятся и потеряют рейтинг. Поэтому главный камень пошел на хитрость. Он сказал камням, что их рейтинг может увеличиваться от посещения новых площадок. Для полной правдоподобности он составил карту площадок. Поскольку главный камень ленивый, он не стал рисовать все пути между площадками, а просто нарисовал минимальное количество проходов между площадками, чтобы из любой площадки можно было добраться в любую другую. Для каждого прохода камень рассчитал, чему равно время перекачивания по нему. Дело осталось за малым — отправить камней в путь. Но камни не дураки и все ещё припоминают главному камню прошлое скатывание. Сначала камни вообще никуда не хотели катиться, но немного подумав они поняли, что некоторые скатывания могут увеличить их рейтинг, а ведь рейтинг это главное в жизни любого камня.

Камни начали дружно собираться в путь, но тут они вспомнили, что у них нет правил изменения рейтинга. Для составления правил они пошли к главному камню, который объявил камням следующие правила: Исходно рейтинг каждого камня обнуляется. Далее, камни могут катиться между площадками, и зарабатывать себе новый рейтинг. За посещение любой площадки к рейтингу камня прибавляется рейтинг площадки. Но за все хорошее нужно платить. Главный камень постановил, что за каждую минуту катания надо платить одной единицей рейтинга. А чтобы камни не злоупотребляли увеличением рейтинга, любому камню разрешено посещать каждую из площадок не более одного раза.

Камни очень расстроились от новых правил, как всегда все хорошие инициативы сверху закончились обнулением. Теперь камни сидят в депрессии и думают, что сделать со своим нулевым рейтингом и как его увеличить. Главный камень, оставаясь на позитиве, напомнил остальным камням, что от движения между площадками их рейтинг увеличится, и покатился по своему маршруту зарабатывать рейтинг. Воодушевившись поступком главного камня, остальные камни сплотились и решили последовать его примеру и тоже начать катиться. К сожалению, считать камни не умеют, а поэтому они решили попробовать все варианты маршрутов и выявить из них самый выгодный.

Однако камни забыли о самом страшном. Ведь во время движения по маршруту между какими-то площадками, у камней рейтинг может уменьшаться. И если у какого-то камня рейтинг становится отрицательным, то камень начинает настолько сильно горевать, что погибает на месте, так и не докатившись до конца маршрута.

Вы, являясь чуткий человеком, не можете пройти мимо горя камней и уж тем более допустить их смерти. Помогите камням и посчитайте количество маршрутов между различными площадками, таких что камень сможет прокатиться по маршруту до конца живым, и его итоговый рейтинг увеличится.

Формат входных данных

В первой строке дано единственное целое число n ($1 \leq n \leq 100\,000$) — число площадок.

В следующей строке дано n чисел $r_1, r_2, r_3, \dots, r_n$ ($1 \leq d_i \leq 10^9$) — рейтинг, который зарабатывает камень за посещение каждой из площадок.

В следующих $n - 1$ строках даны по три числа a_i, b_i и t_i ($1 \leq a_i, b_i \leq n, 1 \leq c_i \leq 10^9$) — номера площадок, которые соединены i -й проход, и время перекачивания по нему.

Формат выходных данных

В единственной строке выведите количество маршрутов между различными парами площадок, таких что они соответствуют правилам главного камня, и при перекачивании по маршруту камень

докатится живым, а его рейтинг в конце маршрута станет больше, чем был до начала движения.

Примеры

стандартный ввод	стандартный вывод
2 1 3 1 2 2	1
4 1 2 3 4 1 2 1 1 3 2 1 4 3	8
5 3 2 1 1 2 1 2 3 1 3 3 2 4 2 3 5 2	5

Замечание

В первом тесте из условия можно добраться только от площадки 2 до площадки 1. Рейтинг камня увеличится сначала на 3, потом уменьшится на 2, потом увеличится на 1, то есть в итоге увеличится на 2. Если же начать катиться от площадки 1 до площадки 2, то после минуты движения рейтинг камня станет нулевым, а значит при продолжении движения рейтинг опустится ниже нуля, и камень погибнет от горя.

Во втором тесте из условия возможны пути: $1 \rightarrow 2$, $2 \rightarrow 1$, $2 \rightarrow 3$, $3 \rightarrow 1$, $3 \rightarrow 2$, $4 \rightarrow 1$, $4 \rightarrow 2$, $4 \rightarrow 3$.

[А] 3. Бинарная ДНК

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Неизвестный науке вирус охватил человечество, проникая в мозг и подавляя интеллект. Заражение проходило почти бессимптомно. На ранних стадиях наблюдалось лишь чрезмерное использование нейросетей даже для решения самых простых задач. Из-за этого вирус активно прогрессировал. После заражения большей части населения лучшие умы, которые вирус обошёл стороной, выяснили, что он изменяет некоторые участки ДНК, заменяя их на нули и единицы.

Олег работает в лаборатории по изучению мутировавших ДНК. В лабораторию поступила ДНК, которую можно описать строкой s длины n , состоящей из символов «0», «1» и «?».

- Если $s_i = 0$ или 1 , то i -й элемент ДНК мутировал.
- Если $s_i = ?$, то i -й элемент ДНК мутировал не до конца.

Олег разработал алгоритм, однозначно сопоставляющий каждой бинарной строке t положительной длины уникальную задачу по программированию. Он выяснил, что человек с ДНК s может решить задачу, соответствующую строке t , только если из s можно заменить все символы «?» на «0» или «1», а затем удалить произвольные символы (не меняя порядок оставшихся) так, чтобы в результате получилась строка t .

Интеллектом человека с ДНК s он назвал количество задач, которые тот может решить. Иногда ДНК может мутировать, поэтому Олег передал вам q запросов:

- Элемент s_{p_i} мутировал и заменился на c_i ($c_i \in \{0, 1, ?\}$).

После каждой мутации найдите *интеллект*. Поскольку он может быть очень большим, выведите его по модулю 998244353.

Формат входных данных

Первая строка содержит два целых числа n и q ($1 \leq n, q \leq 10^5$) — длина ДНК и количество мутаций соответственно.

Вторая строка содержит строку s длины n ($s_i \in \{0, 1, ?\}$) — элементы ДНК.

Следующие q строк описывают мутации ДНК. i -я из них содержит p_i и c_i ($1 \leq p_i \leq n$, $c_i \in \{0, 1, ?\}$) — описание i -й мутации.

Формат выходных данных

После каждой мутации выведите *интеллект* по модулю 998244353.

Пример

стандартный ввод	стандартный вывод
5 6	28
10?01	22
4 ?	13
1 0	11
3 0	12
4 1	5
5 0	
4 0	

Замечание

После первой мутации ДНК описывается строкой «10??1». Такой человек может решить задачи, соответствующие строкам «10011», «1111», «101», но не задачи «00001», «0000». Обратите внимание, что здесь перечислены не все задачи, которые он может решить.

[А] 4. Задача двух фишек

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Однажды Коперник разгадал тайну Трисоляриса и поспешил к царю Чжоу, чтобы скорее рассказать ему об открытии и спасти цивилизацию. Однако царь Чжоу решил для начала проверить ум Коперника и для этого сыграть с ним в игру.

Чжоу нарисовал на земле ориентированный граф из n вершин и m рёбер. В двух различных вершинах графа лежат две фишки. Далее игра происходит по ходам, на каждом ходу царь выбирает одну из двух фишек, после чего Коперник должен передвинуть эту фишку по одному из исходящих рёбер, но так, что бы две фишки не оказались в одной вершине. Т. е. если выбранная фишка стоит в вершине u , из которой есть ребро в вершину v , и в вершине v не стоит вторая фишка, то Коперник может передвинуть выбранную фишку в вершину v . Если Коперник не может сделать ход, то он проигрывает. Если же Коперник всегда может сделать ход, как бы ни ходил царь Чжоу, то он выигрывает.

Коперник и Чжоу хотят сыграть q игр, в i -й игре фишки изначально расположены в вершинах a_i и b_i . Однако совсем скоро может наступить эра Хаоса, поэтому у них нет времени играть столько партий. Поэтому они просят вас для каждой игры узнать, кто из них выигрывает.

Формат входных данных

В первой строке вводятся два целых числа n и m — количество вершин и рёбер в графе ($2 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5$). Далее в m строках вводятся через пробел пары чисел u_i, v_i , обозначающие, что в графе есть ориентированное ребро из u_i в v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$). В следующей строке вводится целое число q — количество игр ($1 \leq q \leq 10^5$). Далее в q строках вводятся через пробел пары чисел a_i, b_i — начальные положения фишек в играх ($1 \leq a_i, b_i \leq n, a_i \neq b_i$).

Формат выходных данных

Для каждой игры выведите в отдельной строке «Copernicus», если выигрывает Коперник (т. е. у Коперника есть стратегия, позволяющая ему сделать любое количество ходов, вне зависимости от игры Чжоу), иначе выведите «King Zhou».

Примеры

стандартный ввод	стандартный вывод
2 1 1 2 1 1 2	King Zhou
9 10 4 9 2 5 5 4 4 3 2 7 1 7 6 5 8 1 9 6 7 8 4 2 5 2 7 2 3 5 9	Copernicus Copernicus King Zhou King Zhou

[А] 5. Миссия: Каблук

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Ателье «Queenswoman» — легендарный дом высокой обувной моды, о котором знают лишь избранные. Его двери открываются только по личному приглашению, а готовые изделия никогда не попадают на витрины обычных магазинов. Каждая пара создаётся вручную мастерами, чьи фамилии известны в лучших кругах Лондона и Парижа, и подбирается с той же точностью, с какой агент мирового уровня выбирает костюм для секретной миссии.

В просторном зале «Queenswoman» на длинной витрине выложены новые туфли, изготовленные по индивидуальным заказам. Каждая модель существует только в единственном экземпляре пары — ровно два одинаковых туфля. Всего n моделей, а значит, на витрине расположено ровно $2n$ туфель, выстроенных в некотором порядке слева направо.

С первого взгляда модели почти невозможно отличить: все они выполнены в одном стиле, из одинаковой кожи, в тех же цветах и с безупречным блеском. Чтобы точно определить, совпадают ли два туфля, нужно поднять их и взглянуть на внутреннюю сторону язычка, где мастер аккуратно указывает характеристики: длину колодки, ширину в носке, ширину в пятке, высоту подъёма, форму мыска, угол изгиба подошвы, толщину стельки, жёсткость задника, высоту каблука, материал подкладки и множество других параметров, которые способны отличить одну пару от другой.

Роксана — главный специалист по подбору обуви в «Queenswoman». Стоит ей один раз заглянуть на язычок, и она запоминает все параметры этой модели навсегда. Благодаря феноменальной памяти Роксана всегда собирает пары без ошибок и лишних движений. Её работа подчиняется строгому порядку: если она уже встречала оба туфля одной модели и они всё ещё на витрине, она сразу убирает их в коробку. В противном случае она идёт слева направо, поднимает первый неразобранный туфель, изучает его характеристики и либо сразу берёт его пару (если помнит, где она находится), либо поднимает следующий туфель и проверяет совпадение. Каждый её ход — это момент, когда она берёт один туфель, затем второй, и либо убирает их с витрины, либо возвращает обратно, если параметры не совпадают. Порядок её действий всегда однозначно определяется текущим расположением обуви.

Но спокойному процессу Роксаны мешает Эдгар — обаятельный клиент с репутацией шутника. Время от времени он меняет местами два туфля на витрине, и Роксане приходится действовать уже с новой раскладкой.

Вам даны изначальное расположение обуви и последовательность таких перестановок. Необходимо сначала определить количество ходов, которое потребуется Роксане для исходной раскладки, а затем, после каждой перестановки, количество ходов для новой раскладки.

Формат входных данных

Первая строка содержит два целых числа n и q — количество моделей обуви и количество перестановок ($1 \leq n, q \leq 200\,000$).

Вторая строка содержит $2n$ целых чисел: i -е число обозначает модель туфля, стоящего на позиции i ($1 \leq a_i \leq n$). Каждая модель встречается ровно дважды.

Далее следуют q строк, каждая из которых содержит два числа ℓ и r ($1 \leq \ell < r \leq 2n$) — позиции туфель, которые Эдгар меняет местами. Все перестановки выполняются последовательно, одна за другой.

Формат выходных данных

В первой строке выведите количество ходов, необходимых для исходной раскладки. В следующих q строках выведите количество ходов после каждой из q перестановок.

Примеры

стандартный ввод	стандартный вывод
3 2 2 1 3 2 3 1 2 4 3 6	5 4 3
4 3 1 1 4 4 3 2 3 2 3 6 2 4 5 8	5 6 7 6

Замечание

Для первого примера начальная последовательность: [2, 1, 3, 2, 3, 1]. Последовательность ходов следующая.

1. Готовых пар нет. Открываются позиции 1 (модель 2) и 2 (модель 1) — модели разные, кладём обратно.
2. Готовых пар нет. Открываются позиции 3 (модель 3) и 4 (модель 2) — модели разные, кладём обратно.
3. Пара модели 2 (позиции 1 и 4) полностью известна и обе туфли на витрине — снимаем их.
4. Готовых пар нет. Открывается позиция 5 (модель 3), и оказывается, что пара этой модели (позиции 3 и 5) полностью известна — снимаем их.
5. Готовых пар нет. Открывается позиция 6 (модель 1), и оказывается, что пара этой модели (позиции 2 и 6) полностью известна — снимаем их.

[А] 6. Игра в шахматы

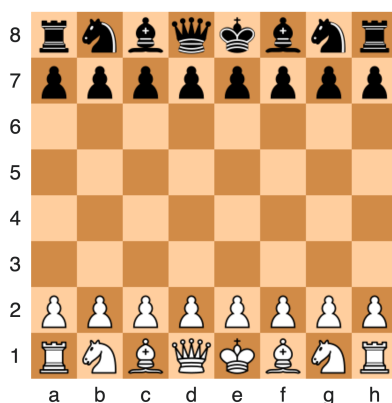
Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Лена очень любит игру в шахматы, и она очень расстроилась, что в этом году у нее не выйдет сыграть в них с Шаи-Хулудом. Это не беда, ведь есть ещё много других людей, с которыми можно играть в шахматы!

Правила шахмат

В шахматы играют на доске, поделённой на равные квадратные клетки, или поля. Размер доски — 8×8 клеток. Вертикальные ряды полей (вертикали) обозначаются латинскими буквами от **a** до **h** слева направо, горизонтальные ряды (горизонтали) — цифрами от 1 до 8 снизу вверх; каждое поле обозначается сочетанием соответствующих буквы и цифры. Поля раскрашены в тёмный и светлый цвета (и называются, соответственно, чёрными и белыми) так, что соседние по вертикали и горизонтали поля раскрашены в разные цвета. Доска располагается так, чтобы ближнее угловое поле справа от игрока было белым (для белых это поле *h1*, для чёрных — поле *a8*).

У игроков в начале игры имеется по одинаковому набору фигур. Фигуры одного из игроков условно называются «белыми», другого — «чёрными». Белые фигуры окрашены в светлый цвет, чёрные — в тёмный. Сами игроки называются «белые» и «чёрные» по цвету своих фигур.



В каждый комплект фигур входят: король, ферзь, две ладьи, два слона, два коня и восемь пешек. В начальной позиции фигуры обеих сторон размещаются так, как показано на диаграмме выше. Белые занимают первую и вторую горизонтали, чёрные — седьмую и восьмую. Пешки расположены на второй и седьмой горизонталях соответственно.

Игра заключается в том, что игроки поочерёдно делают ходы. Первый ход делают белые. За исключением *особых ходов* (взятие на проходе и рокировка, описание данных ходов доступно ниже), ход заключается в том, что игрок перемещает одну из своих фигур на другое поле по следующим правилам:

- Фигуры (кроме коня) передвигаются по прямой линии, при этом все промежуточные поля между начальным и конечным должны быть свободны (на них не должно быть своих или чужих фигур). Исключением является ход коня.
- Ход на поле, занятое своей фигурой, невозможен.
- При ходе на поле, занятое чужой фигурой, она снимается с доски (такой ход называется взятием).
- Король ходит на соседнюю клетку по вертикали, горизонтали или диагонали.
- Ферзь ходит на любое расстояние по вертикали, горизонтали или диагонали.

- Ладья ходит на любое расстояние по вертикали или горизонтали.
- Слон ходит на любое расстояние по диагонали.
- Конь двигается на две клетки по вертикали и затем на одну клетку по горизонтали, или наоборот, на две клетки по горизонтали и на одну клетку по вертикали, тем самым движение коня напоминает заглавную букву «Г» кириллического или заглавную букву «L» латинского алфавитов. Другими словами, конь ходит на одно из полей, ближайших к тому, на котором он стоит, но не на той же самой горизонтали, вертикали или диагонали. При этом цвет поля, на которое ходит конь, всегда противоположен цвету поля, на котором он стоит. Конь отличается от остальных фигур также тем, что другие фигуры, стоящие на пути коня, не препятствуют его ходу (он «перепрыгивает» через них).
- Пешка может ходить только вперёд (направлением «вперёд» называется направление к восьмой горизонтали для белых или к первой для чёрных): без взятия — на одно поле вперёд по вертикали, а со взятием — по диагонали на одно поле вперёд-вправо или вперёд-влево. Если пешка находится на начальном поле (вторая горизонталь для белых и седьмая для чёрных), то она также может сделать ход без взятия на два поля вперёд. Когда пешка ходит на последнюю горизонталь (для белых — на восьмую, для чёрных — на первую), она заменяется по выбору игрока на любую другую фигуру того же цвета, кроме короля и пешки (то есть пешка может превратиться в ферзя, ладью, слона или коня). Такой ход называется *превращением пешки*. Превращение пешки является частью того хода, которым она перемещается на последнюю горизонталь. Таким образом, если, например, превращённая из пешки фигура угрожает королю противника, то этот король немедленно оказывается под шахом. Нет никаких дополнительных ограничений на фигуру, в которую пешка может превратиться в результате превращения (то есть например на доске может стать больше одного ферзя).

Поле называется *битым*, если при своём ходе фигура могла бы взять находящуюся на этом поле фигуру противника (независимо от того, есть ли такая фигура на этом поле). Поле считается битым, даже если фактически ход фигурой туда невозможен, так как ставит под бой собственного короля.

Дополнительно существует два вида *особых ходов*:

- *Рокировка* — если король и одна из ладей того же цвета не двигались с начала игры, то король и эта ладья могут в один ход одновременно сменить положение (рокироваться). При рокировке король сдвигается на две клетки по направлению к ладье, а ладья ставится на поле между начальной и конечной позицией короля. Рокировка невозможна, если король или соответствующая ладья уже ходили. Рокировка временно невозможна, если между королём и ладьёй находится какая-либо фигура, а также если поле, на котором стоит король, или поле, которое он должен пересечь, или поле, которое он должен занять, находится под ударом одной из фигур противника. Обратите внимание, что при шахе король не может сделать рокировку, даже если это перенесет его на небитое поле. Также обратите внимание, что поле, на котором исходно стояла ладья или поле, через которое пройдет ладья но на которое не встанет король может быть битым.
- *Взятие на проходе* — когда пешка совершает ход на две клетки через битое поле, находящееся под ударом пешки противника, то ответным ходом она может быть взята этой пешкой противника. При этом пешка противника перемещается на битое поле, а взятая пешка снимается с доски (пример показан на диаграмме). Взятие на проходе возможно только пешкой непосредственно в ответ на ход пешки через битое поле, на следующих ходах оно уже не разрешено.

Король, находящийся на битом поле, называется *«стоящим под шахом»*. Сделать ход, после которого король противника оказывается под шахом, значит *«объявить шах»*. **Ходы, после которых король сделавшего ход остаётся или оказывается под шахом, запрещены.** Игрок, король которого находится под шахом, обязан немедленно его устранить.

Если король игрока находится под шахом и игрок не имеет ни одного хода, позволяющего устранить этот шах, этот игрок называется *получившим мат* и, соответственно, он терпит поражение. Цель игры и состоит в том, чтобы поставить мат королю противника.

Если игрок при своей очереди хода не имеет возможности сделать ни одного хода по правилам, но король игрока не находится под шахом, такая ситуация называется «*пат*».

Игра завершается выигрышем одной из сторон или ничьей. Ничья фиксируется в следующих случаях (**правила ничьи ниже незначительно отличаются от классических правил шахмат, прочтите их внимательно**):

- Поставлен *пат*.
- Троекратное повторение позиции (не обязательно в течение трёх ходов подряд), причём в понятие позиции в этом случае входит только расположение фигур (данное определение отличается от классических правил шахмат).
- *Правило 50 ходов* — в течение 100 последних полуходов (ходов одного из игроков) не было сделано ни одного взятия и ни одного хода пешкой.

Мат имеет приоритет перед ничьей, то есть если после хода игрока второй игрок находится в матовом положении (но не патовом), то игра завершается победой игрока, сделавшего ход, даже если какое-то из условий ничьи выполнено.

К сожалению, описанные выше классические «шахматы 8×8 » уже не вызывают сильных эмоций, поэтому Лена решила сыграть в усложнённую версию этой игры: «шахматы $m \times n$ ». Однако соперникам Лены не понравились принятые ею обозначения размерности, поэтому они принципиально сели сбоку от доски. Теперь белые пешки ходят по вертикали, а черные — по горизонтали. Это не беда, ведь главное, что черные теперь играют в «шахматы $n \times m$ ».

Лена отказывается признавать доску размером $n \times m$, поэтому она пытается всячески саботировать игру. Попад в очень трудную позицию с обозначениями размеров доски, Лена вынудила соперников применить любимый тактический приём шахматистов. Конечно же, им пришлось ударить кулаком по доске. Но до того, как они это сделают, им надо оценить возможный урон, поэтому они обратились к вам за помощью.

Позицию, в которой находится игра, можно описать матрицей a размера $n \times m$ (начиная с этого момента условия, мы перейдем на сторону соперников Лены, а поэтому размерности у нас будут $n \times m$). Шахматы необычные, поэтому в некоторых клетках этой доски могут стоять гвозди. Таким образом:

- Если $a_{i,j} = «*»$, то в клетке (i, j) находится гвоздь.
- Иначе, $a_{i,j}$ — это какая-то цифра, равная *ценности* фигуры, которая находится в клетке (i, j) . Если $a_{i,j} = 0$, то в этой клетке нет фигуры.

Кулак соперников Лены, которым может быть нанесён удар, может принимать форму произвольного прямоугольника. Таким образом, ударом по доске они могут уничтожить все фигуры в клетках (i, j) , таких, что $l_x \leq i \leq r_x$ и $l_y \leq j \leq r_y$ для каких-то $1 \leq l_x \leq r_x \leq n$ и $1 \leq l_y \leq r_y \leq m$. Если в одной из таких клеток находится гвоздь, соперникам Лены будет очень больно. Несмотря на то, что боль от подобных ударов не настолько сильна, как боль от ограничений $m \times n$, соперники Лены не рассматривают возможность нанести удар по гвоздям.

Пусть S — сумма «ценностей» всех фигур, которые будут уничтожены ударом. *Силу* такого удара определим как S^2 . Соперники Лены просят вас посчитать суммарную *силу* всех возможных ударов, которые они могут нанести, не повредив руку. Помогите им найти эту величину по модулю 998244353.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 2000$) — размеры доски.

Далее следуют n строк. i -я из них содержит m символов $a_{i,1}, a_{i,2}, \dots, a_{i,m}$ ($a_{i,j} = «*»$ или $0 \leq a_{i,j} \leq 9$) — описание позиции на доске.

Формат выходных данных

Выведите одно число: суммарную *силу* всех потенциальных ударов по модулю 998244353.

Примеры

стандартный ввод	стандартный вывод
2 3 1*2 34*	95
4 3 012 345 678 901	18000

Замечание

В первом примере можно нанести удар шестью разными способами. *Силы* таких ударов равны 1^2 , 2^2 , 3^2 , 4^2 , $(1 + 3)^2$ и $(3 + 4)^2$. Таким образом, суммарная сила этих ударов равна $1 + 4 + 9 + 16 + 16 + 49 = 95$.