Задача А. Задача каменного века

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

Когда-то давно Миша и Миша решили придумать интересную задачу на очередной этап РОИ (редкая олимпиада по информатике). Один из них придумал прототип задачи, но другой своровал идею и предложил задачу на другой этап этой же олимпиады. С тех пор первый Миша ждал возможности предложить оригинальную идею на какую-либо другую олимпиаду... Ждал Миша до этого момента!

Вам дан массив a из n целых чисел. Также даны q запросов двух типов:

- Заменить i-й элемент массива на число x.
- Заменить каждый элемент массива на число x.

После выполнения каждого запроса вы должны вычислить сумму всех элементов в массиве.

Формат входных данных

Первая строка содержит два целых числа n и q $(1 \leqslant n, q \leqslant 2 \cdot 10^5)$ — количество элементов в массиве и количество запросов, соответственно.

Вторая строка содержит n целых чисел $a_1, \ldots, a_n$ $(1 \leqslant a_i \leqslant 10^9)$ — элементы массива a.

Каждая из следующих q строк содержит описание очередного запроса. Описание запроса начинается с целого числа t $(t \in \{1, 2\})$, которое обозначает тип запроса:

- Если $t = 1$, далее следуют два целых числа i и x $(1 \leqslant i \leqslant n,\ 1 \leqslant x \leqslant 10^9)$ — позиция элемента, который нужно изменить, и его новое значение.
- Если $t = 2$, далее следует целое число x $(1 \leqslant x \leqslant 10^9)$ — число, на которое нужно заменить каждый элемент массива.

Формат выходных данных

Выведите q целых чисел, каждое в отдельной строке. В i-й строке нужно вывести сумму всех элементов массива после выполнения первых i запросов.

Пример

стандартный ввод	стандартный вывод
5 5 1 2 3 4 5 1 1 5 2 10 1 5 11 1 4 1 2 1	19 50 51 42 5

Замечание

Рассмотрим массив из примера и результат выполнения каждого запроса:

1. Изначально массив равен $[1, 2, 3, 4, 5]$.
2. После выполнения первого запроса массив равен $[5, 2, 3, 4, 5]$. Сумма всех элементов равна 19.
3. После выполнения второго запроса массив равен $[10, 10, 10, 10, 10]$. Сумма всех элементов равна 50.

4. После выполнения третьего запроса массив равен $[10, 10, 10, 10, 11]$. Сумма всех элементов равна 51.
5. После выполнения четвертого запроса массив равен $[10, 10, 10, 1, 11]$. Сумма всех элементов равна 42.
6. После выполнения пятого запроса массив равен $[1, 1, 1, 1, 1]$. Сумма всех элементов равна 5.

Задача B. Шахматы и пути

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

Миша и Костя играют с шахматной доской размера $n \times m$. Строки доски пронумерованы сверху вниз целыми числами от 1 до n, а столбцы — слева направо целыми числами от 1 до m. Таким образом, клетка, находящаяся на пересечении x-й строки и y-го столбца, имеет координаты (x, y).

Клетки поля покрашены в белый и черный цвета шахматной раскраской, а именно: клетка с координатами (x, y) покрашена в белый цвет, если $x + y$ четно, и в черный цвет иначе.

Игра происходит следующим образом: Миша и Костя размещают свои фишки на клетки с координатами (x_1, y_1) и (x_2, y_2) соответственно. Цель Миши — передвинуть свою фишку в клетку (x_3, y_3), а цель Кости — передвинуть свою фишку в клетку (x_4, y_4).

Друзья могут перемещать свои фишки **только** по тем клеткам, которые окрашены в белый цвет. За один ход разрешается передвинуть фишку из клетки (x, y) в одну из четырех клеток: $(x - 1, y)$, $(x, y - 1)$, $(x + 1, y)$ или $(x, y + 1)$. Разумеется, фишку нельзя перемещать за пределы шахматной доски.

Очевидно, что при исходной раскраске доски друзья далеко не всегда смогут достичь своих целей. Поэтому было решено, что за одну операцию Миша и Костя могут перекрасить любую черную клетку в белый цвет. Помогите Мише и Косте вычислить минимальное количество клеток, которые нужно перекрасить в белый цвет, чтобы достичь своих целей. Обратите внимание, что если некоторая фишка изначально находится на клетке, окрашенной в черный цвет, друзьям придется ее перекрасить.

Формат входных данных

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

Вторая строка содержит четыре целых числа x_1, y_1, x_3 и y_3 — координаты начальной и конечной клеток фишки Миши.

Третья строка содержит четыре целых числа x_2, y_2, x_4 и y_4 — координаты начальной и конечной клеток фишки Кости.

Формат выходных данных

Выведите одно целое число — минимальное количество клеток, которые необходимо перекрасить, чтобы игроки смогли переместить свои фишки в требуемые позиции.

Пример

стандартный ввод	стандартный вывод
3 5 1 1 3 5 3 1 1 5	3

Замечание

В первом примере можно, например, перекрасить клетки с координатами $(2, 1)$, $(2, 3)$ и $(2, 5)$.

Задача C. Меховые подпоследовательности

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

Недавно Миша устал от бесконечной прокрастинации и захотел решить какую-нибудь задачу. К сожалению, он неправильно прочитал первую же найденную им задачу, а ошибку заметил уже после того, как отправил решение на проверку. К счастью, изначальная задача оказалась несложной, и Миша все же решил ее. А теперь вам предстоит решить его задачу!

Вам дан массив из n неотрицательных целых чисел. Ваша задача — посчитать количество последовательностей $i_1, i_2, \ldots, i_k$, таких что $1 \leqslant i_1 < \ldots < i_k \leqslant n$, а также для каждого $1 \leqslant j \leqslant k$ выполнено неравенство $|a_{i_j} - M| \leqslant 1$, где $M = \operatorname{mex}\{a_{i_1}, \ldots, a_{i_k}\}$.

Напомним, что $\operatorname{mex}\{a_1, \ldots, a_n\}$ — это минимальное целое неотрицательное число, которое отсутствует в множестве чисел $\{a_1, \ldots, a_n\}$. Например, $\operatorname{mex}\{0, 1, 2\} = 3$, а $\operatorname{mex}\{2, 3\} = 0$.

Формат входных данных

Первая строка содержит единственное целое число n ($1 \leqslant n \leqslant 10^5$) — количество элементов в массиве.

Вторая строка содержит n целых неотрицательных чисел $a_1, \ldots, a_n$ ($0 \leqslant a_i \leqslant 10^9$) — элементы массива a.

Формат выходных данных

Выведите одно целое число — количество искомых последовательностей $i_1, i_2, \ldots, i_k$.

Так как ответ может быть достаточно большим, выведите остаток от деления ответа на число 998 244 353.

Пример

стандартный ввод	стандартный вывод
4 0 2 3 2	4

Замечание

Рассмотрим пример из условия. В нем подходят следующие последовательности индексов:

1. $[1]$. Это множество $\{0\}$, его mex равен 1.
2. $[1, 2]$. Это множество $\{0, 2\}$, его mex равен 1.
3. $[1, 4]$. Это множество $\{0, 2\}$, его mex равен 1.
4. $[1, 2, 4]$. Это множество $\{0, 2\}$, его mex равен 1.

Задача D. Граф? А... А я думал, барон...

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

Однажды Маша прогуливалась по парку и нашла под деревом граф... Удивлены? Вы думали, что в этой задаче будет логичная обоснованная история? Не дождетесь! Так вот...

У Маши есть ориентированный граф, в i-й вершине которого записано некоторое целое положительное число a_i. Изначально Маша может положить в некоторую вершину графа монетку. За один ход разрешается переместить монетку, которая сейчас находится в некоторой вершине u, в любую вершину v, такую что в графе существует ориентированное ребро $u \to v$. Каждый раз, когда монетка оказывается в какой-либо вершине i, Маша записывает в блокнот число a_i (в частности, когда Маша изначально кладет монетку в некоторую вершину графа, она пишет в блокнот число, записанное в этой вершине). Маша хочет сделать ровно $k-1$ ходов таким образом, чтобы максимальное число, записанное ей в блокноте, было как можно меньше.

Формат входных данных

Первая строка содержит три целых числа n, m и k ($1 \leqslant n \leqslant 2 \cdot 10^5$, $0 \leqslant m \leqslant 2 \cdot 10^5$, $1 \leqslant k \leqslant 10^{18}$) — количество вершин и ребер в графе, а также количество ходов, которое хочет сделать Маша, соответственно.

Вторая строка содержит n целых чисел a_i ($1 \leqslant a_i \leqslant 10^9$) — числа, записанные в вершинах графа.

Каждая из следующих m строк содержит два целых числа u и v ($1 \leqslant u \neq v \leqslant n$) — это означает, что в графе существует ребро $u \to v$.

Гарантируется, что граф не содержит петель и кратных ребер.

Формат выходных данных

Выведите одно целое число — минимальное значение максимального числа, которое Маша выписала в блокнот, при оптимальном перемещении монетки.

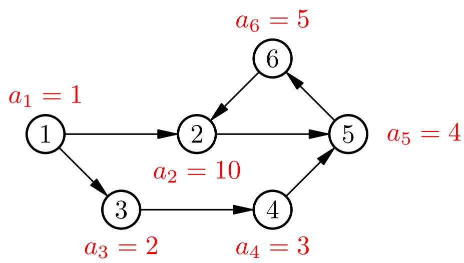
В случае, если Маше не удастся переместить монетку $k-1$ раз, в качестве ответа выведите число -1.

Примеры

стандартный ввод	стандартный вывод
6 7 4 1 10 2 3 4 5 1 2 1 3 3 4 4 5 5 6 6 2 2 5	4
6 7 100 1 10 2 3 4 5 1 2 1 3 3 4 4 5 5 6 6 2 2 5	10
2 1 5 1 1 1 2	-1
1 0 1 1000000000	1000000000

Замечание

На изображении ниже приведен граф, описанный в первых двух примерах.



В первом примере можно выбрать в качестве изначальной вершину 1. После этого можно выполнить три хода: $1 \to 3$, $3 \to 4$ и $4 \to 5$. В итоге в блокнот будут записаны числа $1, 2, 3$ и 4.

Во втором примере можно выбрать в качестве изначальной вершину 2. После этого можно выполнить 99 ходов: $2 \to 5$, $5 \to 6$, $6 \to 2$, $2 \to 5$, и так далее. В итоге в блокнот будут записаны числа $10, 4, 5, 10, 4, 5, \ldots, 10, 4, 5, 10$.

В третьем примере на имеющемся графе не удастся выполнить 4 хода.

Задача E. Типичная вечеринка в общаге

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

Сегодня в общаге праздник — приехал Олег, в честь чего девочки подарили ему строку. Олегу очень понравился подарок, поэтому он тут же придумал и предложил вам, лучшему его другу, следующую задачу.

Вам дана строка s длины n, которая состоит из первых 17 строчных букв латинского алфавита a, b, c, …, p, q и знаков вопроса. А также q запросов. Каждый запрос характеризуется набором попарно различных строчных первых 17 букв латинского алфавита, которые можно использовать чтобы заменить знаки вопроса в строке s.

Ответом на запрос является сумма количества различных подстрок, которые являются палиндромами, по всем строкам, которые можно получить из изначальной строки s путем замены знаков вопроса на разрешенные символы. Ответ необходимо посчитать по модулю 998 244 353.

Обратите внимание! Две подстроки являются различными, когда отличаются их позиции начала и окончания в строке. Т. е. количество различных подстрок, которые являются палиндромами, для строки `aba` будет 4: `a`, `b`, `a`, `aba`.

Рассмотрим примеры замены знаков вопроса на буквы. Например, из строки `aba??ee` при запросе a, b можно получить строки `ababaee` или `abaaaee` но нельзя получить строки `pizza`, `abaee`, `abacaba`, `aba?fee`, `aba47ee`, или `abatree`.

Напомним, что палиндромом называется строка, которая одинаково читается как слева направо, так и справа налево.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leqslant n \leqslant 1\,000$) — длина строки s.

Вторая строка содержит строку s, которая состоит из n строчных букв латинского алфавита и знаков вопроса. Гарантируется, что все буквы в строке принадлежат множеству a, b, c, …, p, q.

Третья строка содержит одно целое число q ($1 \leqslant q \leqslant 2 \cdot 10^5$) — количество запросов.

Далее следуют q строк в каждой из которых содержится единственная строка t — набор символов, которыми можно заменять знаки вопроса ($1 \leqslant |t| \leqslant 17$). Гарантируется, что все буквы в строке принадлежат множеству a, b, c, …, p, q и встречаются не более одного раза.

Формат выходных данных

На каждый запрос выведите одно число — суммарное количество подстрок-палиндромов во всех возможных строках, которые можно получить из строки s, по модулю 998 244 353.

Примеры

стандартный ввод	стандартный вывод
7 ab??aba 8 a b ab abc abcd abcde abcdef abcdefg	14 13 55 105 171 253 351 465
11 ??????????? 6 abcdefghijklmnopq ecpnkhbmlidgfjao olehfan codef glhf q	900057460 712815817 839861037 756843750 70840320 66

Замечание

Рассмотрим первый пример и первый запрос в нём. У нас может получится только одна строка по итогам замены знаков вопроса — `abaaaba`. В ней есть такие подстроки-палиндромы:

1. `a` — подстрока [1; 1].
2. `b` — подстрока [2; 2].
3. `a` — подстрока [3; 3].
4. `a` — подстрока [4; 4].
5. `a` — подстрока [5; 5].
6. `b` — подстрока [6; 6].
7. `a` — подстрока [7; 7].
8. `aa` — подстрока [3; 4].
9. `aa` — подстрока [4; 5].
10. `aba` — подстрока [1; 3].
11. `aaa` — подстрока [3; 5].
12. `aba` — подстрока [5; 7].
13. `baaab` — подстрока [2; 6].
14. `abaaaba` — подстрока [1; 7].

В третьем запросе у нас может получится 4 строки: `abaaaba`, `abababa`, `abbaaba`, `abbbaba`.

Задача F. Маткульт-привет!

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

> Маткульт-привет!
>
> Алексей Савватеев

Сегодня на очередном занятии в математическом кружке, посвященном теории чисел, Сережа узнал много новых для него интересных функций. В частности, ему очень понравилась функция $\varphi(n)$, которая определяется следующим образом: $\varphi(n)$ равно количеству натуральных чисел, не превосходящих n, *взаимно-простых* с n. Эта функция показалась Сереже очень красивой, так как на занятии он узнал несколько ее замечательных свойств. Например, для любых *взаимно-простых* чисел a и b верно, что $\varphi(a \cdot b) = \varphi(a) \cdot \varphi(b)$.

Напомним, что натуральные числа a и b называются *взаимно-простыми*, если их наибольший общий делитель равен единице. Например, числа 5 и 8 являются взаимно-простыми, а числа 12 и 9 — нет (их наибольший общий делитель равен 3).

Приведем некоторые примеры значений функции $\varphi(n)$:

- $\varphi(5) = 4$ (натуральные числа, не превосходящие 5, взаимно-простые с 5: $1, 2, 3, 4$),
- $\varphi(1) = 1$ (существует всего одно натуральное число, не превосходящее 1 — само число 1),
- $\varphi(6) = 2$ (натуральные числа, не превосходящие 6, взаимно-простые с 6: $1, 5$).

Сережа очень любит натуральные числа из промежутка $[l, r]$, то есть числа $l, l+1, \ldots, r$. Начинающему математику тут же захотелось исследовать поведение функции $\varphi(n)$ на промежутке $[l, r]$.

Сережа хочет найти такое натуральное число x, что $l \leqslant x \leqslant r$, а также $\varphi(x) \geqslant \varphi(y)$ для любого натурального числа $l \leqslant y \leqslant r$. Так как Сережа является начинающим математиком, он не справился с этой задачей, поэтому решить ее придется вам.

Формат входных данных

Единственная строка содержит два натуральных числа l и r ($1 \leqslant l \leqslant r \leqslant 10^{12}$).

Формат выходных данных

Выведите одно натуральное число x, для которого верно, что $l \leqslant x \leqslant r$, а также $\varphi(x) \geqslant \varphi(y)$ для любого натурального числа $l \leqslant y \leqslant r$.

Если существует несколько подходящих чисел x, выведите любое из них.

Примеры

стандартный ввод	стандартный вывод
1 6	5
10 10	10
14 16	16

Замечание

В первом примере значения функции $\varphi(n)$ для всех натуральных чисел из промежутка $[1, 6]$ равны: $\varphi(1) = 1, \varphi(2) = 1, \varphi(3) = 2, \varphi(4) = 2, \varphi(5) = 4, \varphi(6) = 2$.

Во втором примере 10 — единственное натуральное число из промежутка $[10, 10]$.

В третьем примере можно вывести в качестве ответа числа 15 или 16, так как $\varphi(14) = 6$, а $\varphi(15) = \varphi(16) = 8$.

Задача G. Спасение Минотавра

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

Лабиринт Минотавра представляет из себя прямоугольную таблицу, состоящую из n строк и m столбцов. Строки пронумерованы целыми числами от 1 до n сверху вниз, а столбцы — целыми числами от 1 до m слева направо. В каждой клетке лабиринта находится либо стена, либо пустое пространство.

Вы выведали у Дедала секретную информацию — про каждую диагональ таблицы вам известна четность количества стен на этой диагонали.

Для спасения Минотавра вам нужно построить любой лабиринт с заданной четностью количества стен на каждой диагонали. Дерзайте!

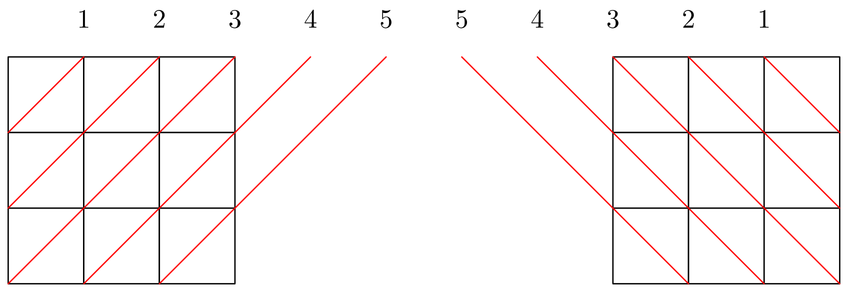
Формат входных данных

Первая строка содержит два целых числа n и m ($2 \leqslant n, m \leqslant 10^5$) — количество строк и столбцов в лабиринте, соответственно.

Вторая строка содержит $n + m - 1$ целых чисел a_i — четность количества стен на диагоналях, идущих вправо-вверх.

Третья строка содержит $n + m - 1$ целых чисел b_i — четность количества стен на диагоналях, идущих вправо-вниз.

Диагонали обоих типов нумеруются начиная с угла верхней строки таблицы. Для лучшего понимания нумерации диагоналей обратите внимание на изображение ниже:



Формат выходных данных

В случае, если подходящего лабиринта не существует, в качестве ответа выведите число -1.

В противном случае в первой строке выведите число k ($0 \leqslant k \leqslant 10^6$) — количество клеток лабиринта, в которых находится стена.

В каждой из следующих k строк выведите через пробел два целых числа r_i и c_i — номер строки и столбца, на пересечении которых находится i-я стена лабиринта.

Все клетки (r_i, c_i) должны быть попарно различны. Если существует несколько решений, выведите любое из них.

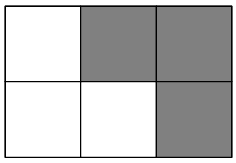
Можно показать, что при данных ограничениях, если существует какое-либо решение, то существует и решение, в котором количество стен не превышает 10^6. Обратите внимание, что от вас **не требуется** минимизировать количество стен в лабиринте.

Примеры

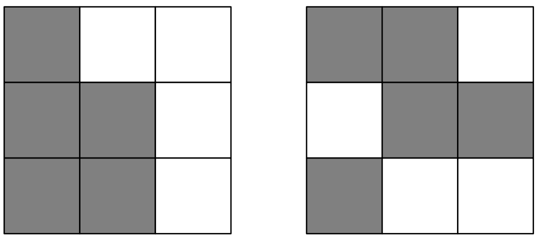
стандартный ввод	стандартный вывод
2 3 0 1 1 1 1 0 0 0	3 1 2 1 3 2 3
2 2 0 1 0 1 0 1	-1
3 3 0 0 0 0 0 0 0 0 0 0	0
3 3 1 1 0 1 0 0 0 0 0 1	5 1 1 2 1 2 2 3 1 3 2

Замечание

На изображении ниже приведено решение для первого примера. Серым цветом обозначены клетки, в которых находятся стены.



В четвертом примере существует два подходящих лабиринта. Оба варианта изображены ниже:



Задача H. Счастливый порядок

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

Рустам занимается изучением натуральных чисел. В какой-то момент его заинтересовали *счастливые* числа. *Счастливыми* Рустам называет натуральные числа, десятичная запись которых состоит только из цифр 4 и 7.

Мальчик начал исследовать различные способы упорядочить эти числа. Он решил, что хочет упорядочить все счастливые числа *лексикографически*.

Так как счастливых чисел бесконечно много, Рустам просит вас помочь ему с исследованием и найти для него n-е счастливое число в лексикографическом порядке.

Напомним, что строка a *лексикографически меньше* строки b, если выполнено одно из двух условий:

1. Строка a является префиксом строки b (и при этом $a \neq b$).
2. Существует такое i $(1 \leqslant i \leqslant \min(|a|, |b|))$, что $a_i < b_i$, и для любого j $(1 \leqslant j < i)$ $a_j = b_j$.

Число x лексикографически меньше числа y, если десятичная запись числа x лексикографически меньше десятичной записи числа y. Для определенности будем считать, что десятичная запись любого числа не содержит ведущих нулей.

Формат входных данных

В единственной строке содержится целое число n $(1 \leqslant n \leqslant 10^6)$.

Формат выходных данных

Выведите n-е счастливое число в лексикографическом порядке.

Пример

стандартный ввод	стандартный вывод
1	4

Задача I. Формализм ради формализма

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

Юра, будучи математиком, уже настолько преисполнился в своем познании, что он уже сто триллионов миллиардов лет решает формальные задачи, в которых нет ни капли легенды. Эта задача именно такая!

Рассмотрим все целые неотрицательные числа из промежутка $[0, 10^n)$. Для удобства дополним все числа ведущими нулями таким образом, чтобы любое число из данного промежутка состояло ровно из n десятичных цифр.

Пусть имеется также некоторый набор пар (u_i, v_i), где u_i и v_i — различные цифры от 0 до 9.

Рассмотрим число x, состоящее из n цифр. Пронумеруем цифры слева направо и обозначим их как $d_1, d_2, \ldots, d_n$. За одну операцию разрешается поменять местами цифры d_i и d_{i+1} тогда и только тогда, когда существует такая пара (u_j, v_j), что верно хотя бы одно из условий:

1. $d_i = u_j$ и $d_{i+1} = v_j$,
2. $d_i = v_j$ и $d_{i+1} = u_j$.

Назовем числа x и y, состоящие из n цифр, *эквивалентными*, если из числа x можно получить число y, пользуясь только операциями, описанными выше. В частности, любое число считается эквивалентным самому себе.

Дано число n и набор из m пар цифр (u_i, v_i). Найдите максимальное k, такое что существует множество чисел $x_1, x_2, \ldots, x_k$ $(0 \leqslant x_i < 10^n)$, для которого верно, что для любых $1 \leqslant i < j \leqslant k$ число x_i **не** эквивалентно числу x_j.

Формат входных данных

Первая строка содержит целое число n $(1 \leqslant n \leqslant 50\,000)$ — количество цифр в рассматриваемых числах.

Вторая строка содержит целое число m $(0 \leqslant m \leqslant 45)$ — количество пар цифр.

Каждая из следующих m строк содержит две цифры u_i и v_i, записанные через пробел $(0 \leqslant u_i < v_i \leqslant 9)$.

Гарантируется, что все пары цифр различны.

Формат выходных данных

Выведите одно число — максимальное число k, такое что существует множество чисел $x_1, x_2, \ldots, x_k$ $(0 \leqslant x_i < 10^n)$, для которого верно, что для любых $1 \leqslant i < j \leqslant k$ число x_i **не** эквивалентно числу x_j.

Так как ответ может быть достаточно большим, выведите остаток от деления числа k на $998\,244\,353$.

Примеры

стандартный ввод	стандартный вывод
1 0	10
2 1 0 1	99
2 9 0 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9	91

Замечание

В первом примере можно составить множество, состоящее из всех чисел от 0 до 9, так как никакие два числа не являются эквивалентными друг другу.

Во втором примере существует единственная пара эквивалентных чисел: 01 и 10. В качестве ответа можно, например, взять множество всех чисел от 0 до 99, кроме числа 1.

Задача J. Ладьи-защитники

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

У вас есть квадратная шахматная доска размера $n \times n$. Строки пронумерованы сверху вниз числами от 1 до n, а столбцы — слева направо числами от 1 до n. Таким образом, каждая клетка доски задается парой целых чисел (x, y) $(1 \leqslant x, y \leqslant n)$, где x — номер строки, а y — номер столбца.

От вас требуется выполнить q запросов трех типов:

- Поставить новую ладью в клетку (x, y).
- Убрать ладью из клетки (x, y), куда ранее была поставлена ладья.
- Проверить, верно ли, что каждая клетка *подпрямоугольника* $(x_1, y_1) - (x_2, y_2)$ доски атакована хотя бы одной ладьей.

Подпрямоугольником называется множество клеток доски (x, y), для которых верно, что $x_1 \leqslant x \leqslant x_2$ и $y_1 \leqslant y \leqslant y_2$.

Напомним, что клетка (a, b) атакована ладьей, находящейся в клетке (c, d), если $a = c$ или $b = d$. В частности, клетка, в которой находится ладья, атакована этой ладьей.

Формат входных данных

Первая строка содержит два целых числа n и q $(1 \leqslant n \leqslant 10^5,\ 1 \leqslant q \leqslant 2 \cdot 10^5)$ — размер доски и также количество запросов, соответственно.

Каждая из следующих q строк содержит в себе описание очередного запроса. Описание запроса начинается с целого числа t $(t \in \{1, 2, 3\})$, которое обозначает тип запроса:

- Если $t = 1$, далее следуют два целых числа x и y $(1 \leqslant x, y \leqslant n)$ — координаты клетки, на которую нужно поставить новую ладью. Гарантируется, что в момент данного запроса в клетке (x, y) не стоит ладья.
- Если $t = 2$, далее следуют два целых числа x и y $(1 \leqslant x, y \leqslant n)$ — координаты клетки, с которой нужно убрать ладью. Гарантируется, что в момент данного запроса в клетке (x, y) стоит ладья.
- Если $t = 3$, далее следуют четыре целых числа x_1, y_1, x_2 и y_2 $(1 \leqslant x_1 \leqslant x_2 \leqslant n,$ $1 \leqslant y_1 \leqslant y_2 \leqslant n)$ — подпрямоугольник, для которого нужно проверить, верно ли, что каждая его клетка атакована хотя бы одной ладьей.

Гарантируется, что среди q запросов есть хотя бы один запрос третьего типа.

Формат выходных данных

Для каждого запроса третьего типа в отдельной строке выведите «`Yes`» (без кавычек), если каждая клетка данного подпрямоугольника атакована хотя бы одной ладьей.

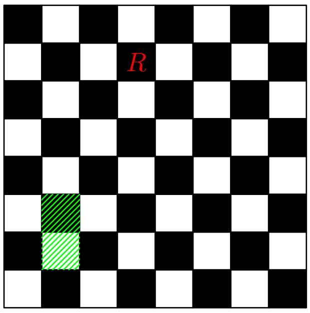
В противном случае выведите «`No`» (без кавычек).

Пример

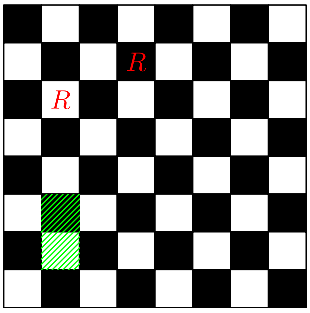
стандартный ввод	стандартный вывод
8 10 1 2 4 3 6 2 7 2 1 3 2 3 6 2 7 2 1 4 3 3 2 6 4 8 2 4 3 3 2 6 4 8 1 4 8 3 2 6 4 8	No Yes Yes No Yes

Замечание

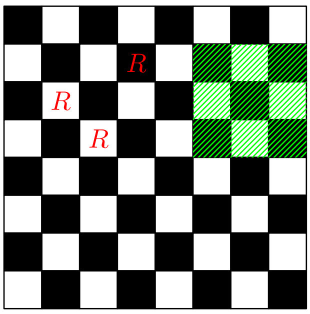
Рассмотрим пример. После первых двух запросов доска будет выглядеть следующим образом (буквой R обозначены клетки, в которых находится ладья, а зеленым выделен прямоугольник запроса третьего типа):



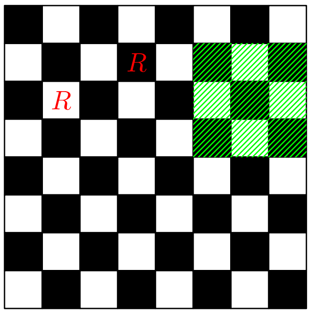
Доска после выполнения третьего и четвертого запросов:



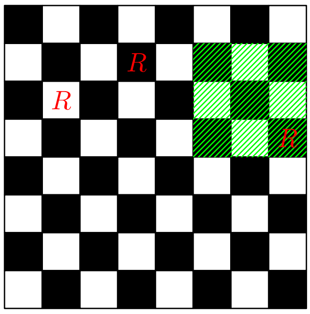
Доска после выполнения пятого и шестого запросов:



Доска после выполнения седьмого и восьмого запросов:



Доска после выполнения последних двух запросов:



Задача K. Фатальная ошибка

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

На дворе был теплый майский вечер, Захар только что дописал свою дипломную работу, название которой настолько сложно, что может испугать неподготовленного читателя. Студент, обрадовавшись, решил распечатать написанный на компьютере текст на n прямоугольных листах бумаги размера $h \times w$ сантиметров. После того, как дипломная работа была распечатана, Захар сложил все листы бумаги в аккуратную стопку: наверху стопки лежала первая страница текста, под ней — вторая, и так далее.

Радости Захара не было предела до тех пор, пока он не совершил фатальную ошибку, решив попить чай и, конечно же, разлив весь напиток прямиком на стопку бумаги. В результате этого на каждом листе бумаги образовалось пятно, имеющее по счастливой случайности форму выпуклого многоугольника. Пятно было одинаково хорошо заметно с каждой стороны каждого листа бумаги. Более того, так как чай медленно распространялся по стопке сверху вниз, пятно, образовавшееся на i-м сверху листе, оказалось *строго вложено* в пятно, образовавшееся на $(i-1)$-м сверху листе.

Будем говорить, что многоугольник A *строго вложен* в многоугольник B, если ломаные, образующие границы многоугольников, не имеют ни одной общей точки, а также любая вершина многоугольника A находится внутри многоугольника B.

Захар был в бешенстве. Он схватил все n листов бумаги и яростно разбросал их по комнате. Немного успокоившись, студент решил собрать все листы обратно в стопку, но оказалось, что он забыл проставить номера страниц на листах, и теперь не знает, в каком порядке они шли изначально. Более того, некоторые листы во время полета по комнате неоднократно перевернулись.

Более формально, каждый лист бумаги при падении на пол мог либо оказаться в исходном состоянии, либо повернуться на 180 градусов таким образом, что текст теперь напечатан на листе не сверху вниз, а наоборот, либо перевернуться наизнанку, либо и перевернуться наизнанку, и повернуться на 180 градусов одновременно.

Захар тут же решил посчитать, сколько существует различных способов собрать стопку бумаги таким образом, чтобы пятно на i-м сверху листе оказалось строго вложено в пятно на $(i-1)$-м сверху листе. Перед тем, как собрать стопку, Захар может повернуть и/или перевернуть каждый лист бумаги так, чтобы он оказался в одном из четырех описанных выше состояний.

Два способа собрать стопку считаются различными, если выполнено одно из двух условий:

1. Порядок следования листов в двух способах различается.

2. Порядок следования листов в двух способах совпадает, но есть хотя бы один лист, который находится в разных состояниях в первом и втором способе.

Формат входных данных

Первая строка содержит три целых числа n, w и h ($1 \leqslant n \leqslant 10^5$, $1 \leqslant w < h \leqslant 10^9$) — количество листов бумаги и их размеры, соответственно.

Следующие строки содержат описания пятен, образовавшихся на каждом из n листов бумаги.

Первая строка описания каждого пятна содержит целое число k_i ($3 \leqslant k_i \leqslant 10^5$) — количество вершин в многоугольнике, задающем пятно, образовавшееся на i-м листе бумаги.

Каждая из следующих k_i строк содержит два целых числа x_{ij} и y_{ij} ($0 \leqslant x_{ij} \leqslant w$, $0 \leqslant y_{ij} \leqslant h$) — координаты вершин очередного многоугольника в порядке обхода против часовой стрелки. Будем считать, что начало координат каждого листа бумаги расположено в его левом нижнем углу.

Гарантируется, что все многоугольники являются выпуклыми, и никакие три вершины одного многоугольника не лежат на одной прямой.

Также гарантируется, что суммарное количество вершин всех многоугольников не превосходит 10^5, то есть $\sum\limits_{i=1}^{n} k_i \leqslant 10^5$.

Формат выходных данных

Выведите одно целое число — количество различных способов собрать стопку бумаги таким образом, чтобы пятно на i-м сверху листе оказалось строго вложено в пятно на $(i-1)$-м сверху листе.

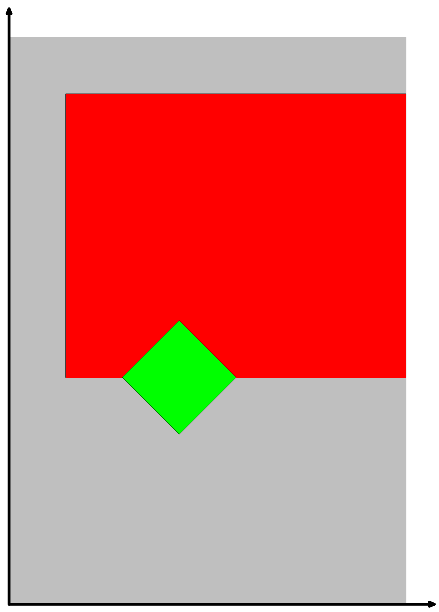
Так как ответ может быть достаточно большим, выведите остаток от деления ответа на число 998 244 353.

Пример

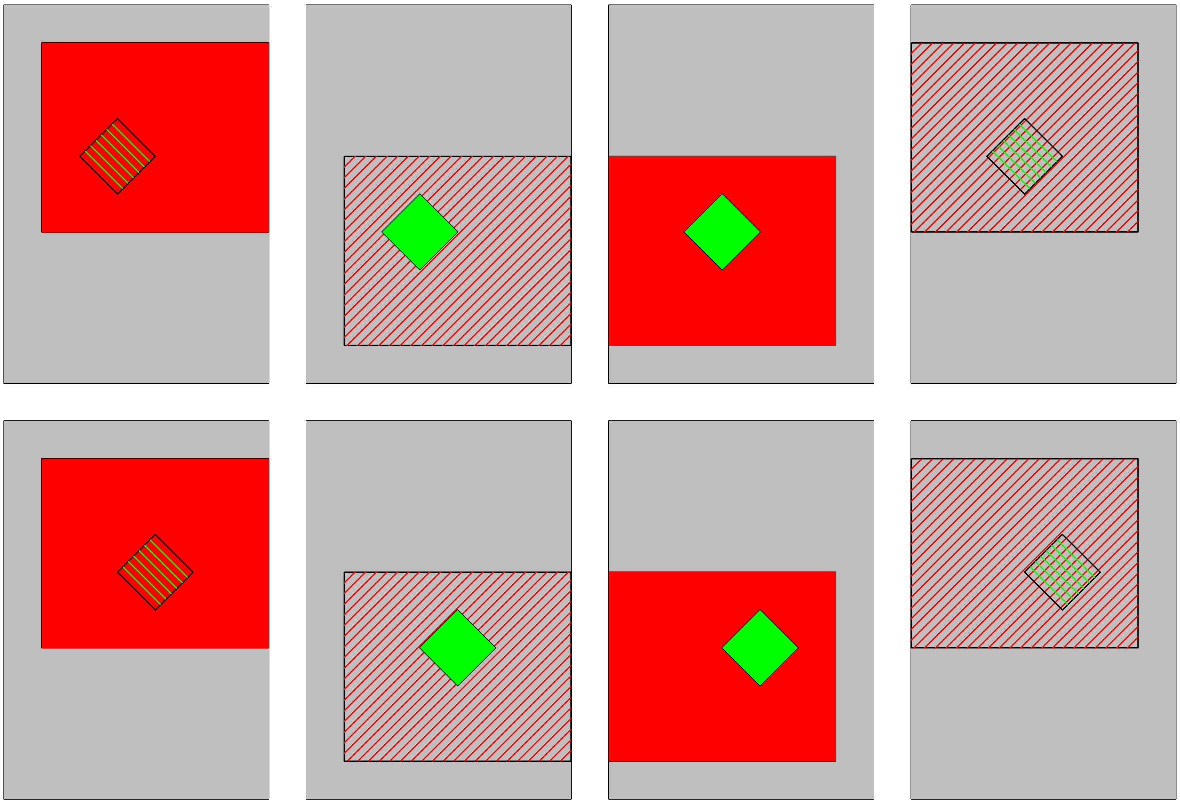
стандартный ввод	стандартный вывод
2 7 10 4 1 4 7 4 7 9 1 9 4 3 5 2 4 3 3 4 4	8

Замечание

Рассмотрим первый пример. На рисунке ниже изображены два многоугольника в их изначальных состояниях. Для удобства восприятия многоугольники наложены друг на друга. Первый многоугольник обозначен красным цветом, а второй — зеленым цветом. Лист бумаги обозначен серым цветом.



На рисунке ниже изображены восемь способов сложить стопку из данных двух листов. Если какой-то из многоугольников заштрихован, то это означает, что лист с данным многоугольником перевернут наизнанку.



Задача L. AvtoBus

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

Наступила весна, и руководство автобусного парка AvtoBus дало распоряжение поменять у всех автобусов зимние шины на летние.

Вы владеете небольшой фирмой, занимающейся обслуживанием автобусов, и только что вам поступил долгожданный заказ на замену n шин. Вы знаете, что автобусный парк владеет автобусами двух типов: с двумя осями (у таких автобусов 4 колеса) и с тремя осями (у таких автобусов 6 колес).

Вы не знаете, сколькими автобусами какого типа владеет парк AvtoBus, поэтому вам стало интересно, сколько автобусов может быть в парке. Вам нужно определить, какое минимальное и какое максимальное количество автобусов может быть в парке, зная, что суммарное количество колес у всех автобусов равно n.

Формат входных данных

В единственной строке записано целое число n ($1 \leqslant n \leqslant 10^{18}$) — суммарное количество колес у всех автобусов.

Формат выходных данных

В единственной строке выведите два целых числа x и y ($1 \leqslant x \leqslant y$) — минимальное и максимальное возможное количество автобусов, которыми владеет парк.

В случае, если для данного n не существует подходящего количества автобусов, выведите в качестве ответа число -1.

Примеры

стандартный ввод	стандартный вывод
4	1 1
7	-1
24	4 6
998244353998244352	166374058999707392 249561088499561088

Замечание

В первом примере известно, что суммарное количество колес у всех автобусов равно 4. Это значит, что в автобусном парке есть ровно один автобус с двумя осями.

Во втором примере не трудно показать, что не существует такого количества автобусов, чтобы их суммарное количество колес было равно 7.

В третьем примере суммарное количество колес у всех автобусов равно 24. Возможны следующие варианты:

- Четыре автобуса с тремя осями.
- Три автобуса с двумя осями и два автобуса с тремя осями.
- Шесть автобусов с двумя осями.

Таким образом, минимальное количество автобусов равно 4, а максимальное — 6.