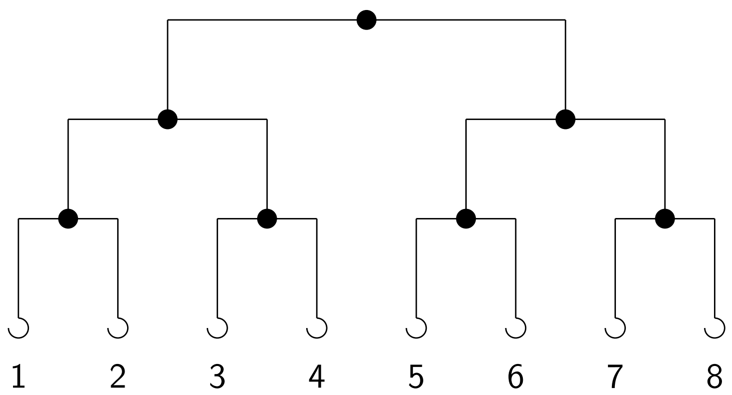
Инновационная вешалка

Инновационная вешалка состоит из n уровней, состоящих из связанных между собой стержней. Уровень i (при $i \in \{0, 1, \ldots, n-1\}$) состоит из 2^i горизонтальных стержней. Середина стержня на уровне 0 прикреплена к стене. На всех остальных уровнях середина j-ого (при $j \in 1, \ldots, 2^i$) стержня прикреплена к левой части $\lceil \frac{j}{2} \rceil$-го стержня предыдущего уровня при нечетном j, или к правой части того же стержня при четном j. На обоих концах каждого стержня на последнем уровне висят крючки для одежды. Крючки пронумерованы слева направо числами от 1 до 2^n.

Например, вешалка при $n = 3$ выглядит следующим образом:



Маша хочет повесить все свои куртки на свою новую вешалку. Вес каждой куртки равен единице. Чтобы не сломать хрупкую конструкцию, она должна вешать куртки в таком порядке, чтобы разница между общим весом на левом конце любого стержня и общим весом на правом конце того же стержня после добавления очередной куртки была равна 0 либо 1. (По законам физики разница может быть равна и -1, однако Маша считает перекос в правую сторону ужасным.) Стержни такие тонкие, что их весом можно пренебречь.

Маша наслышана о вашем профессионализме и просит вашей помощи. Напишите программу, которая по заданным n и k находит номер крючка по модулю $10^9 + 7$, на который Маша должна повесить куртку на k-м шаге.

Входные данные

На единственной строке заданы два целых числа n и k.

Выходные данные

Выведите одно целое число — номер крючка, на который Маша должна повесить куртку на k-м шаге по модулю $10^9 + 7$.

Ограничения

- $n \in [1, 10^6]$.
- $k \in [1, \min\{2^n, 10^{18}\}]$.

Подзадачи

- **20 баллов:** $n \in [1, 10]$.
- **20 баллов:** $n \in [1, 20]$.
- **60 баллов:** основные ограничения.

Пример 1

Входные данные

```
3 2
```

Выходные данные

```
5
```

Комментарий

В этом примере крючки должны быть использованы в следующем порядке: 1, 5, 3, 7, 2, 6, 4, 8. На втором шаге Маша должна повесить свою куртку на крючок номер 5.

Пример 2

Входные данные

```
5 10
```

Выходные данные

```
19
```

Комментарий

Здесь порядок использования крючков такой: 1, 17, 9, 25, 5, 21, 13, 29, 3, 19, и т. д.

XORanges

Джеймс настолько любит апельсины, что он сделал для них сканер, используя 4 камеры и компьютер Raspberry Pi 3b+, и начал создавать 3D изображения апельсинов. Его процессор изображений не очень хорош, поэтому в результате сканирования он получает только 32-битное целое число, которое содержит информацию о повреждениях на кожуре. 32-битное число D представляется последовательностью из 32 битов, каждый из которых может быть нулем или единицей. Если занумеровать биты с 0, то можно получить число D, сложив 2^i для каждого i-го бита, равного единице. Более формально, число D задается последовательностью $d_{31}, d_{30}, \dots d_0$, если $D = d_{31} \cdot 2^{31} + d_{30} \cdot 2^{30} + \ldots + d_1 \cdot 2^1 + d_0 \cdot 2^0$. Например, число 13 представляется как $0, \ldots, 0, 1, 1, 0, 1$.

Джеймс отсканировал n апельсинов; тем не менее иногда он решает пересканировать один из апельсинов (i-й апельсин) во время выполнения вашей программы. Это значит, что после пересканирования нужно использовать новое значение для i-го апельсина.

Джеймс хочет анализировать полученные данные. Он считает операцию "исключающего ИЛИ" (XOR) очень интересной, поэтому решает использовать ее в вычислениях. Он выбирает диапазан апельсинов с l до u (где $l \leq u$) и хочет вычислить результат операции XOR, примененной ко всем числам диапазана, ко всем парам соседних элементов диапазона, всем последовательностям из 3 соседних элементов, и т. д. вплоть до последовательности из $u - l + 1$ соседних элементов (всех элементов диапазона).

То есть, если $l = 2$, $u = 4$ и A — массив полученных в результате сканирования значений, программа должна вернуть результат следующего выражения $a_2 \oplus a_3 \oplus a_4 \oplus (a_2 \oplus a_3) \oplus (a_3 \oplus a_4) \oplus (a_2 \oplus a_3 \oplus a_4)$, где $\oplus$ обозначает XOR и a_i обозначает i-й элемент в массиве A.

Операция XOR над двумя числовыми значениями определяется так:

Если i-й бит первого числа такой же, как i-й бит второго, то i-й бит результата равен `0`; если i-й бит первого числа отличается от i-го бита второго числа, то i-й бит результата равен `1`.

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Например, $13 \oplus 23 = 26$.

$13 =$	$0\ldots001101$
$23 =$	$0\ldots010111$
$13 \oplus 23 = 26 =$	$0\ldots011010$

Входные данные

В первой строке входных данных расположены 2 целых положительных числа n и q (общее число операций пересканирования и анализа данных).

В следующей строке расположены n разделенных пробелом целых неотрицательных чисел, которые представляют значения массива A (результаты сканирования апельсинов). Элемент a_i содержит описание i-го апельсина. Индексы нумеруются начиная с 1.

Операции описываются в следующих q строках, с помощью трех разделенных пробелами целых положительных чисел.

Если операция имеет тип 1 (пересканирование), то первое число равно 1, за ним следует i (индекс апельсина, который хочется пересканировать) и j (результат пересканирования i-го апельсина).

Если тип операции 2 (анализ), первое число равно 2, за ним следуют l и u.

Выходные данные

Вы должны вывести в точности одно целое число для каждой операции типа 2 (анализ данных). Вы должны выводить каждое значени в отдельной строке. Обратите внимание, что i-я строка выходных данных должна соответствовать i-й операции типа 2.

Ограничения

- $a_i \leq 10^9$
- $0 < n, q \leq 2 \cdot 10^5$

Подзадачи

1. **[12 баллов]**: $0 < n, q \leq 100$
2. **[18 баллов]**: $0 < n, q \leq 500$ и нет операций изменения значений
3. **[25 баллов]**: $0 < n, q \leq 5000$
4. **[20 баллов]**: $0 < n, q \leq 2 \cdot 10^5$ и нет операций изменения значений
5. **[25 баллов]**: Нет дополнительных ограничений

Примеры

Пример 1

Входные данные

```
3 3
1 2 3
2 1 3
1 1 3
2 1 3
```

Выходные данные

```
2
0
```

Комментарий

В начале $A = [1, 2, 3]$. Первая операция производится над полным диапазоном значений. Результат анализа есть $1 \oplus 2 \oplus 3 \oplus (1 \oplus 2) \oplus (2 \oplus 3) \oplus (1 \oplus 2 \oplus 3) = 2$.

Затем значение первого апельсина меняется на 3. Это приводит к изменению результата в операции анализа всех данных (в диапазоне $[1, 3]$): $3 \oplus 2 \oplus 3 \oplus (1 \oplus 2) \oplus (2 \oplus 3) \oplus (1 \oplus 2 \oplus 3) = 0$.

Пример 2

Входные данные

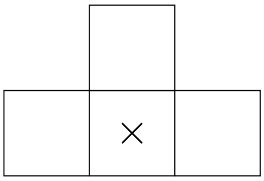
```
5 6
1 2 3 4 5
2 1 3
1 1 3
2 1 5
2 4 4
1 1 1
2 4 4
```

Выходные данные

```
2
5
4
4
```

T - Покрытие

Если вы когда-нибудь играли в Тетрис, вы должны знать, что одна из фигур выглядит так:



Мы будем называть эту фигуру *T-тетрамино*; тетрамино — это просто модное слово для связной геометрической фигуры, состоящей из 4 клеток. Клетка, отмеченная $\times$, будет называться *центральной клеткой*.

Манка рисует прямоугольную сетку с m рядами и n столбцами и записывает некоторое число в каждую клетку. Ряды таблицы пронумерованы от 0 до $m-1$, и столбцы от 0 до $n-1$. Она также отмечает некоторые клетки как *особые*, например, окрашивая их в красный цвет. После этого она просит Нику, свою подругу, расположить несколько T-тетрамино на сетке таким образом, чтобы выполнялись следующие условия:

- Количество T-тетрамино должно быть таким же, как и особых клеток.
- Для каждого T-тетрамино, его центральная клетка должна лежать на особой клетке.
- Ни одна пара T-тетрамино не должна перекрываться.
- Все T-тетрамино должны полностью лежать в нарисованной сетке. Обратите внимание, что существует 4 возможных расположения для каждого T-тетрамино ($\top$, $\perp$, $\vdash$, and $\dashv$).

Если условия не могут быть выполнены, Ника должна ответить `No`; если они выполнимы, она должна найти такое расположение T-тетрамино, чтобы максимизировать сумму чисел в клетках, покрытых T-тетрамино. В этом случае она должна сообщить Манке максимальную сумму покрытых T-тетрамино клеток.

Напишите программу, чтобы помочь Нике решить эту задачу.

Входные данные

Первая строка входных данных содержит два целых числа m и n — высоту и ширину таблицы. Следующие m строк содержат по n целых чисел в интервале $[0, 1000]$. j-е целое число в i-й строке соответствует числу, записанному в j-й клетке i-го ряда сетки.

Следующая строка содержит число k ($k \in \{1, \ldots, mn\}$) — количество особых клеток в таблице. Каждая из последующих строк содержит по два числа r_i и c_i, $r_i \in \{0, \ldots, m-1\}$ and $c_i \in \{0, \ldots, n-1\}$ — номер строки и столбца i-ой особой клетки, соответственно.

Гарантируется, что все координаты особых клеток уникальны.

Выходные данные

Выведите максимально возможную сумму чисел, покрытых Т-тетрамино клеток, либо `No`, если не существует способа расставить их.

Ограничения

$1 \le mn \le 10^6$.

Подзадачи

- **5 баллов:** $k \le 1000$; для любой пары особых клеток i и j, выполняется условие $|r_i - r_j| > 2$ или $|c_i - c_j| > 2$.
- **10 баллов:** $k \le 1000$; любые пары различных особых клеток (r_i, c_i) и (r_j, c_j), для которых $|r_i - r_j| \le 2$ и $|c_i - c_j| \le 2$, являются смежными по стороне, более формально выполняется следующее условие ($|r_i - r_j| = 1$ и $|c_i - c_j| = 0$) или ($|r_i - r_j| = 0$ и $|c_i - c_j| = 1$).
- **10 баллов:** $k \le 1000$; для любой пары различных особых клеток i и j, с $|r_i - r_j| \le 2$ и $|c_i - c_j| \le 2$, выполняется условие $|r_i - r_j| \le 1$ и $|c_i - c_j| \le 1$.
- **10 баллов:** $k \le 1000$; все особые клетки находятся в одном ряду.
- **15 баллов:** $k \le 10$.
- **20 баллов:** $k \le 1000$.
- **30 баллов:** нет дополнительных ограничений.

Пример 1

Входные данные

```
5 6
7 3 8 1 0 9
4 6 2 5 8 3
1 9 7 3 9 5
2 6 8 4 5 7
3 8 2 7 3 6
3
1 1
2 2
3 4
```

Output

```
67
```

Комментарий

Чтобы получить максимальную сумму, Ника должна расположить Т-тетрамино так:

- ⊣ на клетке (1, 1);
- ⊢ на клетке (2, 2);
- ⊥ на клетке (3, 4).

Пример 2

Входные данные

```
5 6
7 3 8 1 0 9
4 6 2 5 8 3
1 9 7 3 9 5
2 6 8 4 5 7
3 8 2 7 3 6
3
1 1
2 2
3 3
```

Выходные данные

```
No
```

Странное Стреляндское Странствие

Европейская юниорская олимпиада по информатике 2542 года проводится в Стреляндии. Стреляндия имеет форму таблицы из m строк (занумерованных числами от 0 до $m-1$) и n столбцов (занумерованных числами от 0 до $n-1$), в каждой клетке которой находится город. Пусть (r, c) обозначает клетку, находящуюся в строке r и столбце c. Общежитие, в котором живут участники, находится в клетке $(0, 0)$, а зал соревнований находится в клетке $(m-1, n-1)$.

Особенность Стреляндии в том, что в некоторых городах находятся огромные стрелки. За одну операцию стрелку можно повернуть на 90 градусов по часовой стрелке. Изначально каждая стрелка указывает на север, восток, юг или запад. Организаторы олимпиады хотят воспользоваться особенностью своей страны для навигации участников.

Участники олимпиады будут следовать по стрелкам. Из каждого города они будут переходить в соседний город по направлению стрелки. Если они окажутся в городе, в котором нет стрелки, или если они выйдут за границы Стреляндии, они заблудятся и не смогут участвовать в олимпиаде. Организаторы соревнований хотят, чтобы все участники благополучно добрались до зала соревнований из общежития (клетка (0, 0)) в зал соревнований (клетка $(m-1, n-1)$). Для этого они могут повернуть какие-то стрелки. Помогите им определить минимальное число описанных выше поворотов, необходимое для достижения цели, или сообщите, что участники не смогут добраться до зала соревнований при любой ориентации стрелок.

Входные данные

Первая строка входных данных содержит два целых числа m и n — число строк и столбцов. Следующие m строк содержат по n символов, задающих изначальное направление стрелок (`N` – север, `E` – восток, `S` – юг, `W` – запад, `X` – нет стрелки). Последний символ последней строки (то есть, символ, соответствующий залу соревнований) всегда будет `X`.

Символ `N` означает направление вверх, `E` означает "вправо", `S` означает "вниз", и `W` означает "влево".

Выходные данные

Выведите минимальное число поворотов стрелок, которые нужно сделать организаторам. Если решения нет, выведите -1.

Ограничения

- $1 \le m \le 500$.
- $1 \le n \le 500$.
- Каждая клетка содержит один из символов: `N`, `E`, `S`, `W`, `X`.

Подзадачи

- **10 баллов:** $m = 1$; каждая клетка содержит либо символ `E` либо символ `X`.
- **12 баллов:** $m = 1$.
- **12 баллов:** $m = n = 3$.
- **16 баллов:** $m = 2$.
- **50 баллов:** нет дополнительных ограничений.

Пример 1

Входные данные

```
3 3
EES
SSW
ESX
```

Выходные данные

```
3
```

Комментарий

В этом примере оптимальное решение — изменить `W` в клетке $(1, 2)$ на `S`, повернув стрелку три раза.

Пример 2

Входные данные

```
3 3
EES
SSW
EEX
```

Выходные данные

```
0
```

Комментарий

В этом примере ничего не нужно делать, участники и так доберутся до зала соревнований.

Пример 3

Входные данные

```
3 4
EXES
WSNS
XNNX
```

Выходные данные

```
4
```

Комментарий

Повернуть стрелку в клетке $(0, 0)$ один раз (чтобы получить `S`), стрелку в клетке $(1, 0)$ два раза (чтобы получить `E`), и стрелку в клетке $(2, 1)$ один раз (чтобы получить `E`).

Башня

Фермер Джереми скучает по вечерам, поэтому он развлекается игрой в простую игру. Он хочет построить башню из чисел, написанных на кусочках бумаги. Он всегда начинает с куска бумаги, на котором написано 1.

Джереми может написать другое число на куске бумаги и разместить его в вершине башни. При этом новое значение должно быть суммой чисел из любого диапазона последовательно расположенных листочков башни. А именно: пусть в текущий момент башня состоит из n кусков бумаги. Джереми выбирает диапазон $[l, u]$, где $1 \leq l \leq u \leq n$, считает сумму чисел в нем, пишет ее на новом куске бумаги и размещает его на вершине башни.

Джереми хочет построить T башен с заданными конечными числами на вершине каждой из башен. Помогите ему найти требуемую последовательность ходов для каждой из башен.

Входные данные

В первой строке входных данных находится положительное целое число T (количество башен, которые Джереми хочет построить с различными значениями на вершинах).

В следующих T строках расположено по одному положительному числу q — значения, которые Джереми хочет получить в вершинах каждой из башен. Строительство каждой из башен производится независимо от других.

Выходные данные

Для каждого значения q

- сначала выведите одно число $0 \leq s \leq 1000$ — требуемое число ходов для получения башни с этим числом на вершине;
- в следующих s строках должны быть расположены по два разделенных пробелом числа l и u — границы диапазона номеров листочков, значения на которых следует сложить на соответствующем ходу.

Ограничения

- $1 \leq T \leq 1000$
- $1 \leq q \leq 10^{18}$

Подзадачи

1. **[1 тест - 10 баллов]**: $T \leq 10$ и $q \leq 10$
2. **[1 тест - 10 баллов]**: $T \leq 20$ и $q \leq 20$
3. **[1 тест - 10 баллов]**: $T = 100$ и $q \leq 100$
4. **[1 тест - 10 баллов]**: $T = 1000$ и $q \leq 10^4$
5. **[1 тест - 10 баллов]**: $T = 1000$ и $q \leq 10^5$
6. **[1 тест - 10 баллов]**: $T = 1000$ и $q \leq 10^6$
7. **[1 тест - 10 баллов]**: $T = 1000$ и $q \leq 10^9$
8. **[1 тест - 10 баллов]**: $T = 1000$ и $q \leq 10^{12}$
9. **[2 теста - 20 баллов]**: нет дополнительных ограничений.

Система оценки

В задаче будет 10 тестов, в каждом нужно построить T башен. Для каждого теста баллы начисляются по следующим правилам:

- если решение получило желаемые значения за минимальное число ходов для всех башен теста, то вы получаете 10 баллов за соответствующий тест;
- в противном случае, ваше решение будет оцениваться как минимальное значение баллов по всем башням теста, которое вычисляется для каждой из башен по формуле $1 + \frac{minimum\ steps}{solution\ steps} \cdot 7$, с округлением до двух десятичных знаков после запятой;
- если решение для одной из башен некорректное, оценка за тест 0 баллов.

Для всех башен существует корректное решение.

Пример

Входные данные

```
3
2
3
7
```

Выходные данные

```
2
1 1
1 2
3
1 1
2 2
1 3
4
1 1
1 2
2 3
1 4
```

Пояснение

В этом примере $T = 3$.

Первое желаемое значение на вершине равно 2, второе 3, третье 7. Изначально каждая башня содержит только кусок бумаги с числом $[1]$.

За первый ход Джереми может выбрать только диапазон $[1, 1]$, поэтому второе число в башне может быть только 1.

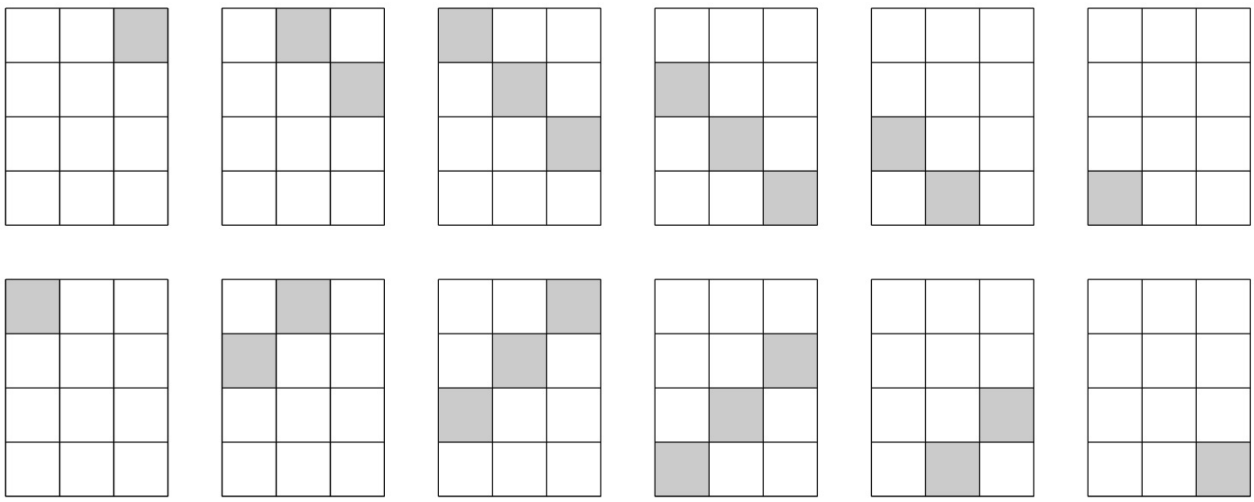
Когда Джереми хочет получить в вершине число2, он должен выбрать диапазон $[1, 2]$ (с 1-е- по 2-е число в башне). Так как их сумма равна 2, это дает желаемый результат.

Второе желаемое число 3. Чтобы получить 3, существует несколько способов (в том числе потому, что на втором шаге можно снова выбрать диапазон $1\ 1$).

Покраска прямоугольника

Сережа хочет покрасить прямоугольную таблицу, состоящую из m строк (пронумерованных от 0 до $m-1$) и n столбцов (пронумерованных от 0 до $n-1$). Изначально все клетки таблицы белые. На каждом шаге он выбирает какую-то диагональ (см. Пример 2) и закрашивает все клетки на этой диагонали в свой любимый цвет. Однако стоимость покраски некоторых диагоналей может быть дороже других, вне зависимости от их длины. По заданной стоимости покраски каждой из диагоналей определите минимальную общую стоимость покраски всех клеток в таблице. Обратите внимание, что клетки можно перекрашивать дважды.

Прямоугольная сетка из m строк и n столбцов имеет $2m+2n-2$ диагоналей. Например, если $m=4$ и $n=3$, то всего существует 12 диагоналей:



Входные данные

Входные данные состоят из трех строк.

Первая строка содержит числа m и n.

Вторая строка содержит $m+n-1$ чисел, которые задают стоимость покраски диагоналей по направлению $\searrow$. i-е число (при $i \in \{1,\ldots,m+n-1\}$) относится к диагонали, в которой разность номера строки и номера столбца равна $i-n$. Таким образом первое число относится к диагонали, состоящей только из одной клетки с координатами $(0,n-1)$ (строка 0, столбец $n-1$), второе число определяет стоимость покраски диагонали включающей в себя клетки $(0,n-2)$ и $(1,n-1)$, и т. д.

Третья строка содержит $m+n-1$ чисел, которые определяют стоимость покраски диагоналей по направлению $\nearrow$. i-е число (при $i \in \{1, \dots, m+n-1\}$) относится к диагонали, в которой сумма номера строки и номера столбца равна $i-1$. Таким образом, первое число относится к диагонали, состоящей только из одной клетки с координатами $(0, 0)$, второе число определяет стоимость покраски диагонали включающей в себя клетки $(1, 0)$ и $(0, 1)$, и т. д.

Выходные данные

Выведите минимальную стоимость покраски всей таблицы.

Ограничения

- $1 \le m \le 2 \cdot 10^5$.
- $1 \le n \le 2 \cdot 10^5$.
- Стоимости покраски — целые числа в интервале $[1, 10^9]$.

Подзадачи

- **10 баллов:** $m, n \le 4$.
- **10 баллов:** $m, n \le 10$.
- **10 баллов:** $m, n \le 20$.
- **20 баллов:** $m, n \le 2000$.
- **10 баллов:** $m = 1$ и $n \le 2 \cdot 10^5$.
- **20 баллов:** $m = n \le 2 \cdot 10^5$.
- **20 баллов:** нет дополнительных ограничений.

Пример 1

Входные данные

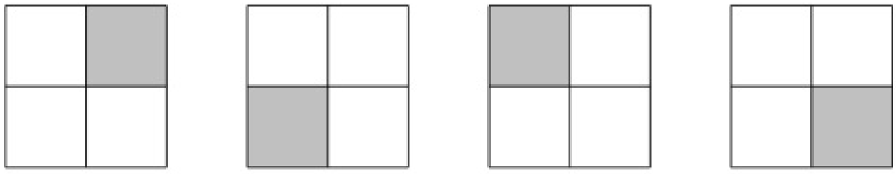
```
2 2
1 3 1
1 3 1
```

Выходные данные

```
4
```

Комментарий

В этом примере для минимизации стоимости должны быть покрашены следующие диагонали:



Покраска каждой из этих диагоналей стоит 1, соответственно итоговая стоимость равна 4.

Пример 2

Входные данные

```
4 3
2 3 9 3 4 3
2 3 3 1 2 4
```

Выходные данные

```
14
```

Комментарий

В этом случае минимальная стоимость получается покраской следующих диагоналей стоимостью 3, 2, 3, 3, 1 и 2 соответственно: