

Задача А. Ограбление банка

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

Мир находится под большой угрозой! Никому не понятно, под какой именно, но чтобы его спасти, Дэдпулу и Росомахе необходимо ограбить банк, а для этого им нужен надежный план.

Известно, что план ограбления банка из n действий можно представить в виде бинарной строки s длины n . Далее план реализуется следующим образом: пока в плане есть более одного действия, строится новый план s^* , в котором $s_i^* = s_i \oplus s_{i+1}$ для всех i от 1 до $|s| - 1$, где $|s|$ — это длина строки s , а \oplus означает операцию «исключающее ИЛИ». Заметьте, что после выполнения очередного шага количество шагов в плане уменьшается на 1.

Считается, что план является успешным, если после его выполнения в плане остается единственный шаг, равный 1. Помогите Дэдпулу и Росомахе понять, является ли план s успешным.

Формат входных данных

В единственной строке задается бинарная строка s — описание изначального плана ($1 \leq |s| \leq 10^6$).

Формат выходных данных

В единственной строке выведите ответ 1, если план является успешным, иначе выведите 0.

Примеры

стандартный ввод	стандартный вывод
1010	0
1101	1

Замечание

В первом примере план будет преобразован следующим образом: $1010 \rightarrow 111 \rightarrow 00 \rightarrow 0$.

Во втором примере план будет преобразован как $1101 \rightarrow 011 \rightarrow 10 \rightarrow 1$.

Задача В. Массивы Росомахи: математическая ловкость и фантазия

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

Уважаемые участники, хочу вам сообщить, что условие этой задачи было написано без использования нейронных сетей. Пожалуйста, приступайте к ее решению самостоятельно. Удачи!

— ChatGPT

Дэдпул и Росомаха соревнуются в создании массивов. На этом связь легенды задачи с фильмом заканчивается, расскажем вам правила соревнования.

В начале Дэдпул называет три целых числа n , l и r . Это означает, что Росомахе необходимо будет построить массив целых чисел длины n , в котором все элементы лежат от l до r включительно. Обозначим $r - l + 1$, то есть длину интервала доступных чисел, за m .

Теперь Росомаха должен построить массив a , удовлетворяющий ограничениям, со следующими свойствами:

- любые три подряд идущих числа a_i , a_{i+1} и a_{i+2} попарно взаимно просты; иными словами, $\gcd(a_i, a_j) = 1$ для любых i и j , для которых $|i - j| \leq 2$;
- среди всех элементов a есть хотя бы $\lfloor \frac{3m}{4} \rfloor - 1$ различных.

Понятно, что это далеко не всегда возможно сделать, поэтому Дэдпул обещает, что для его запросов будет выполняться $n \geq m$.

Помогите Росомахе придумать такой массив, чтобы одержать победу! Гарантируется, что в каждом тесте ответ существует.

Формат входных данных

В единственной строке даны три целых числа n , l и r — длина искомого массива и интервал, в котором должны лежать его значения ($4 \leq n \leq 2 \cdot 10^5$; $1 \leq l \leq r \leq 10^9$; $n \geq r - l + 1 \geq 4$).

Формат выходных данных

В единственной строке выведите через пробел n целых чисел — элементы подходящего массива.

Примеры

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

Задача С. Награда за спасение мультивселенной

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

Дэдпул и Росوماха спасли множество линий времени от полного уничтожения. За это, им, разумеется, полагаются награды (конечно, супергероям платят, и при этом много — иначе как бы они позволили себе свои костюмы и оружие).

Всего им выдали n наград, i -я из которых имеет стоимость a_i . Сначала Логан и Уэйд попробовали разделить награды поровну, то есть так, чтобы каждому досталось множество наград одинаковой суммарной стоимости. Разумеется, у них это сходу не получилось, поэтому после небольшой драки они решили поступить следующим образом:

- для начала выбрать два непересекающихся набора наград одинаковой суммарной стоимости s ; наборы должны быть выбраны таким образом, чтобы s было **максимально**;
- затем отнести все оставшиеся награды в УВИ и убедить их раздвоить каждую из них (забрать вариант той же награды из другой линии времени), чтобы затем можно было и их тоже поделить поровну.

Разумеется, после потасовки они уже не сильно соображают, да и не факт, что план сработает — УВИ бывают довольно упрямыми, так что задача посчитать, награды какой суммарной стоимости в итоге получит каждый из них, если вдруг все пойдет по плану, выпала вам.

Формат входных данных

Первая строка содержит число n — количество наград ($1 \leq n \leq 500$).

Следующие n строк содержат по одному положительному целому числу a_i — стоимости наград. Гарантируется, что $\sum a_i$ не превосходит 10^5 .

Формат выходных данных

Выведите единственное целое число — суммарную стоимость наград, которые каждый заберет домой.

Примеры

стандартный ввод	стандартный вывод
4 2 3 1 6	6
5 2 3 5 8 13	18

Замечание

В первом примере каждый может сразу получить по 6 и в УВИ идти не придется.

Во втором примере каждый может получить по 13, после чего останутся награды суммарной стоимостью 5. Они будут раздвоены, и каждый получит в сумме 18.

Задача D. Милпул и Догпуля

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

Это задача с двойным запуском. На каждом тесте ваше решение будет запущено дважды.

Милпул и Догпуля не в первый раз путешествуют вместе, поэтому у них уже есть любимое развлечение, за которым можно провести скучный вечер. Они играют в следующую игру:

1. Милпул загадывает целое неотрицательное число n , после чего каким-то образом преобразует его в битовую строку (последовательность нулей и единиц) s ;
2. Догпуля убедительно лает на получившуюся битовую строку, из-за чего к ней в конец дописывается какое-то (возможно, нулевое) число произвольных бит;
3. от умиления Милпул забывает, какое число он изначально загадывал, и должен восстановить его по получившейся последовательности q .

Ваша цель — играть за Милпула и выиграть (успешно восстановить исходное число).

Во время **первого запуска** вашему решению на вход подается целое неотрицательное число n от 1 до 10^{18} . Вы должны вывести битовую строку s длины не более $\sqrt{2} \cdot \lceil \log_2 n \rceil + 5$.

При **втором запуске** ваше решение получает на вход строку q длины не более 1000, полученную из строки s на выходе первого запуска дописыванием произвольного числа бит в конец. Ваше решение должно восстановить число n , поданное на вход во время первого запуска.

Формат входных данных

При первом запуске единственная строка ввода содержит число 1 и целое число n ($1 \leq n \leq 10^{18}$).

При втором запуске единственная строка ввода содержит число 2 и строку q , полученную из вывода первого запуска описанным в условии образом ($1 \leq |q| \leq 1000$).

Формат выходных данных

При первом запуске выведите непустую строку s длины не более $\sqrt{2} \cdot \lceil \log_2 n \rceil + 5$, состоящую только из символов '0' и '1' — ваш код числа n .

При втором запуске по полученной строке q восстановите и выведите целое число n , изначально поданное на ввод.

Протокол взаимодействия

Во избежание получения некорректных вердиктов вроде `Idleness Limit Exceeded` или `Security Violation` заканчивайте вывод каждой строки символом перевода строки ('`\n`').

Примеры

	стандартный ввод	стандартный вывод
1	1	
2	010001	010
1	15664	
2	11111111000111	11111111 15664

Замечание

В условии в примерах даны сразу ввод и вывод для обоих запусков. Обратите внимание, что ввод и вывод второго запуска могут отличаться, если ваше решение в первом запуске выводит другую строку.

Задача Е. Потеря медальона

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

Это интерактивная задача.

Перед очередным противостоянием двух непокорных героев — Дэдпула и Росомахи, Росомаха потерял свой старый медальон, дающий невероятные силы владельцу. Как только Логан обнаружил пропажу, ему сразу стало ясно — виноват Дэдпул.

Как любитель бесконечных шуток и игр, Уэйд решил испытать терпение Росомахи, спрятав медальон на координатной прямой в некоторой точке с целой **неотрицательной** координатой (известной только Дэдпулу) и предложив ему такую игру:

За каждый ход Логан может назвать одно натуральное число x . После этого Дэдпул сделает одно из двух действий:

1. Если **текущая** координата, на которой находится медальон, делится на x — Дэдпул просто ответит на запрос числом 1.
2. Если же **текущая** координата, на которой находится медальон, не делится на x , Дэдпул ответит на запрос числом 0, и **сдвинет** медальон ближе к нулю ровно на единицу (более формально — уменьшит координату на 1).

Дэдпул, считая это забавным, отвечает Логану лишь на один вопрос в минуту. Он уверен, что зрителям такое развлечение понравится, а вот Логан больше беспокоится о том, что этот артефакт попадет в руки Кассандры Новы, что случится всего через 52 минуты, и тогда ее будет уже не остановить.

Помогите Росомахе угадать **начальное** местоположение медальона, чтобы спасти мир.

Формат входных данных

Каждый тест содержит несколько наборов входных данных. Первая строка входных данных содержит одно целое число t ($1 \leq t \leq 50$) — количество наборов входных данных.

Гарантируется, что сумма ответов (начальных координат медальона) по всем наборам входных данных не превосходит $5 \cdot 10^5$.

Протокол взаимодействия

Взаимодействие с интерактором проходит в виде запросов со стороны вашей программы и ответов со стороны интерактора. Вы можете не более 52 раз сделать запрос — число x ($1 \leq x \leq 10^9$), на который получите ответ 1 или 0. Чтобы запросить очередное значение, выведите на отдельной строке «? x », где вместо x укажите нужное число. Если вы не превысили лимит в 52 запроса вида «?», в следующей строке интерактор выведет единственное целое число — 0 или 1, по правилу, описанному в условии.

Для того, чтобы вывести ответ на задачу, напечатайте «! c », где вместо c должна быть начальная координата медальона ($0 \leq c \leq 5 \cdot 10^5$). Вывод ответа не учитывается в количестве запросов. После этого вам выведется **вердикт** — 0 или 1 в случае положительного и отрицательного ответа, соответственно.

Если в какой-то момент ваша программа превышает лимит в 52 запроса, ваша программа завершится с вердиктом **WA** (Wrong Answer).

Важно: не забывайте после каждого запроса сбрасывать буфер вывода, чтобы интерактор получил ваш запрос. Это можно сделать с помощью `std::cout.flush()` в C++, `System.out.flush()` в Java и `sys.stdout.flush()` в Python, а также аналогичными командами в других языках. Если ваша программа не сбрасывает буфер вывода, она получит вердикт **TL** (Time Limit) или **IL** (Idleness Limit).

Пример

стандартный ввод	стандартный вывод
1	? 500000
0	? 500000
0	? 500000
0	? 500000
1	? 500000
1	! 3

Задача F. Погоня в Пустоте

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

Логан и Дэдпул пытаются сбежать от Алиота в Пустоте. Алиот, похожий на облако, может занимать произвольное пространство и распространяться в любом направлении по поверхности Пустоты.

Пустота состоит из n локаций, пригодных для жизни (с натяжкой). Герои находятся в локация 1, а вход в укрытие расположен в локация n . Алиот уже проник в k локаций a_1, \dots, a_k , причем известно, что его нет в локация с героями и локация с укрытием.

Также между какими-то локациями есть пути или тоннели — всего их m штук, и i -й из них соединяет локация x_i и y_i . По каждому такому пути можно за одинаковое время переместиться с одной локация на другую в любом направлении.

Ломая четвертую стену, Дэдпул предлагает вам посчитать, сколько максимум тоннелей между локациями можно достроить так, чтобы герои добрались до убежища **строго** раньше, чем их догонит Алиот, если они будут двигаться с одинаковой скоростью.

Если герои и Алиот точно окажутся в одной локация одновременно, выведите, что для них нет спасения. Также учтите, что Алиот именно «распространяется», а не перемещается — при его движении в новую локация, старая локация также остается занятой им.

Формат входных данных

В первой входных данных даны три целых числа n , m и k — число локаций в Пустоте, число построенных между локациями тоннелей и число локаций, занятых Алиотом ($3 \leq n \leq 10^5$; $1 \leq m \leq 2 \cdot 10^5$; $1 \leq k \leq n - 2$).

Во второй строке входных данных перечислены k целых чисел a_i — номера локаций, занятых Алиотом ($2 \leq a_i \leq n - 1$).

В следующих m строках дано описание тоннелей, построенных между локациями. Описание тоннеля состоит из двух чисел x_i и y_i — номеров локаций, которые этот тоннель соединяет ($1 \leq x_i, y_i \leq n$; $x_i \neq y_i$).

Формат выходных данных

Выведите ответ на задачу — наибольшее общее число тоннелей между локациями, при котором герои успеют добраться до убежища, не встретившись с Алиотом.

Если у героев нет шансов не побег вне зависимости от того, сколько тоннелей будет построено к уже имеющимся, вместо этого выведите -1 .

Примеры

стандартный ввод	стандартный вывод
3 1 1 2 1 3	1
7 6 3 5 6 2 1 7 6 2 1 4 3 6 5 3 2 1	12
4 5 1 3 1 4 3 4 4 2 1 3 3 2	-1

Задача G. Ловушка поезда

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

Это интерактивная задача.

Однажды, в один из своих обычных дней (не «обычных», полных путешествий по временным линиям, а обычных), Дэдпул решил отправиться на расслабляющую поездку в поезде — да-да, тот самый поезд, который пересекает всю страну. Не раздумывая, он встал на платформе и сел в первый попавшийся вагон.

Уэйд наслаждался видами из окна. Поездка шла, как ни странно, гладко. Но через несколько часов Дэдпул начал замечать нечто странное: он был уверен, что видел этот же красный амбар за окном не так давно. Немного подумав, он понял, что на самом деле это не обычный день, а «обычный» и УВИ заперли его в циклическом поезде (то есть, в поезде, где можно вернуться к тому же вагону, из которого начался путь, двигаясь в одном направлении). Пытаясь перейти в соседний вагон, он осознал ещё одну проблему — абсолютно все вагоны выглядели **одинаково**, за исключением одной кнопки, которая переключала состояние обогревателя в вагоне. Изначально Дэдпулу ничего не было известно о состояниях обогревателей в вагонах.

Дэдпул понял, что для того, чтобы выбраться из этой ловушки, ему нужно определить количество вагонов в поезде. У него есть следующие действия:

1. «F» — перейти в следующий вагон;
2. «D» — перейти в предыдущий вагон;
3. «C» — потрогать обогреватель в текущем вагоне (чтобы определить его состояние — включен или выключен);
4. «S» — нажать кнопку и поменять состояние обогревателя (то есть, выключить обогреватель, если он включен, и включить, если он выключен).

Помогите Дэдпулу определить количество вагонов в поезде. Поскольку он хочет сделать это как можно быстрее, вы должны потратить **не более** $16 \cdot n$ действий, где n — количество вагонов в поезде.

Формат входных данных

Каждый тест содержит несколько наборов входных данных. Первая строка входных данных содержит одно целое число t — количество наборов входных данных ($1 \leq t \leq 3000$).

Гарантируется, что сумма ответов (количества вагонов) по всем наборам входных данных не превосходит 3000.

Протокол взаимодействия

Взаимодействие с интерактором проходит в виде запросов со стороны вашей программы и ответов со стороны интерактора. Вы можете выполнить одно из действий, описанных в условии, не более $16 \cdot n$ раз. Для некоторых действий вы получите ответы:

1. «F» — перейти в следующий вагон. Ответа от интерактора не последует;
2. «D» — перейти в предыдущий вагон. Ответа от интерактора не последует;
3. «C» — потрогать обогреватель в текущем вагоне. Интерактор выведет 0, если обогреватель выключен, и 1 в противном случае.
4. «S» — нажать кнопку. Ответа от интерактора не последует.

Чтобы вывести ответ на задачу, напечатайте «! n », где вместо n должно быть предложенное вами количество вагонов в поезде ($1 \leq n \leq 3000$). Этот вывод не учитывается в количестве запросов. После этого интерактор выведет вердикт — 1, если ваше предположение верно, и 0 в противном случае (в этом случае ваша программа получит вердикт WA — Wrong Answer).

Если в какой-то момент ваша программа превышает лимит в $16 \cdot n$ запросов, ваша программа завершится с вердиктом WA (Wrong Answer).

Важно: не забывайте после каждого запроса сбрасывать буфер вывода, чтобы интерактор получил ваш запрос. Это можно сделать с помощью `std::cout.flush()` в C++, `System.out.flush()` в Java и `sys.stdout.flush()` в Python, а также аналогичными командами в других языках. Если ваша программа не сбрасывает буфер вывода, она получит вердикт TL (Time Limit) или IL (Idleness Limit).

Пример

стандартный ввод	стандартный вывод
1	C
0	F C
1	F C
0	S C
1	D C
1	D C
1	! 2
1	

Задача Н. Героический поступок

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

В какой-то из вселенных огромной Мультивселенной Marvel для отключения «Потрошителя времени» требовался не настолько героический поступок, как «нарушить энергетический поток материи и антиматерии Потрошителя ценой своей жизни». Достаточно было просто закоротить его.

Чтобы отключить Потрошитель времени, нужно соединить проводами три точки в схеме его материнской платы. Если представить схему как координатную решетку, то эти три места находятся в точках (x_1, y_1) , (x_2, y_2) и (x_3, y_3) с целыми координатами.

Новые провода можно проводить только по линиям решетки (то есть по вертикалям и горизонталям с целыми координатами). Помогите альтернативным Дэдпулу и Росомахе определить минимальную суммарную длину проводов, необходимую, чтобы соединить все три выхода.

Примечание: точки считаются соединенными, если для любых двух из них существует последовательность сегментов проводов, ведущих от одной из них к другой (в частности, не обязательно соединять их тремя непересекающимися проводами «по кругу»).

Формат входных данных

В каждой из трех строк ввода через пробел даны целые числа x_i и y_i — координаты соответствующей точки схемы ($-10^9 \leq x_i, y_i \leq 10^9$).

Формат выходных данных

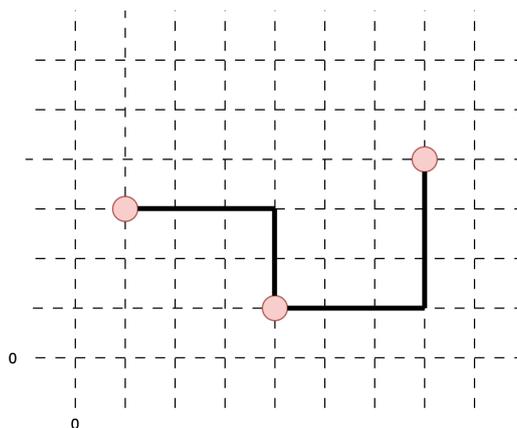
Выведите единственное целое число — минимальную суммарную длину проводов, необходимую для соединения трех данных точек.

Примеры

стандартный ввод	стандартный вывод
0 1 1 0 0 -1	3
1 3 4 1 7 4	9

Замечание

Один из способов соединить точки во втором примере показан на рисунке ниже:



Задача I. Нестабильность времени

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

У УВИ довольно сложная работа: измерять нестабильность временных линий. Дэдпул с Росомой явно прибавили им этой работы, поэтому они пустили в ход новый прибор.

Всего им интересны n временных линий, занумерованных целыми числами от 1 до n . Для m из них известна их *нестабильность* f — целое неотрицательное число. Для остальных известны лишь показания их нового прибора.

Прибор, который мы назовем «Измеритель» (очень изобретательно), выдает результаты следующего вида: «среди временных линий от l -й до r -й включительно линии с номерами x_1, \dots, x_k имеют наиболее высокие уровни нестабильности». То есть для любого i от 1 до k и для любого d от l до r , не совпадающего ни с каким из x_i , $f(x_i) > f(d)$.

По информации обо всех измерениях, определите возможный уровень нестабильности для каждой временной линии, если известно, что он может лежать от 1 до 10^9 включительно. Также может оказаться такое, что имеющиеся данные противоречивы (Измеритель еще не до конца отлажен).

Формат входных данных

Первая строка содержит три целых числа n , s и m — количество временных линий, точно известных уровней нестабильности и измерений, соответственно ($1 \leq s \leq n \leq 10^5$; $1 \leq m \leq 2 \cdot 10^5$).

В i -й из следующих s строк даны два целых числа p_i и d_i , которые означают, что линия времени номер p_i имеет нестабильность ровно d_i ($1 \leq p_i \leq n$; $1 \leq d_i \leq 10^9$). Гарантируется, что $p_i > p_{i-1}$.

Следующие m строк описывают измерения прибора: i -я строка начинается с трех целых чисел l_i , r_i и k_i — границ отрезка i -го измерения и количества наиболее нестабильных линий ($1 \leq l_i < r_i \leq n$; $1 \leq k_i \leq r_i - l_i$). А затем в той же строке следуют k_i целых чисел x_j в порядке возрастания ($l_i \leq x_j \leq r_i$).

Такое измерение показывает, что нестабильность любой линии из множества $\{x_1, \dots, x_{k_i}\}$ строго выше, чем у любой линии из множества $\{l_i, \dots, r_i\} \setminus \{x_1, \dots, x_{k_i}\}$.

Гарантируется, что сумма всех k_i не превышает $2 \cdot 10^5$.

Формат выходных данных

Если измерения некорректны, в единственной строке выведите «NO» (без кавычек).

Иначе, в первой строке выведите «YES», а во второй — последовательность из n чисел, каждое из которых находится в промежутке от 1 до 10^9 — значения нестабильности каждой линии времени.

Если существует несколько верных решений, выведите любой из них.

Пример

стандартный ввод	стандартный вывод
5 2 2	YES
2 7	6 7 1000000000 6 3
5 3	
1 4 2 2 3	
4 5 1 4	

Задача J. Другие

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

Среди членов сопротивления, которых Уйэд называет «Другими», есть строгая иерархия в виде подвешенного дерева. В корне находится лидер сопротивления под номером 1, и у каждого другого члена сопротивления есть один непосредственный начальник, всего n мутантов.

Чтобы организовать сражение с Кассандрой Новой, необходимо для этого сначала оценить, каковы их шансы, и продумать план. Для этого необходимо для нескольких ключевых членов сопротивления (таких, как Электра, Блейд, Гамбит и X-23), определить количество их подчиненных d -го уровня.

Подчиненные d -го уровня в организации определяются следующим образом:

- при $d = 0$ это только сам этот член организации;
- иначе это множество всех **непосредственных** подчиненных у подчиненных $(d - 1)$ -го уровня.

Иными словами, это все потомки данной вершины в дереве на расстоянии ровно d от нее. Ответьте на q запросов описанного вида.

Формат входных данных

В первой строке даны два целых числа n и q — количество членов организации и запросов, соответственно ($1 \leq n \leq 10^5$; $1 \leq q \leq 2 \cdot 10^5$).

В следующей строке даны $n - 1$ целых чисел p_2, \dots, p_n — номера непосредственных начальников членов организации со второго по n -го ($1 \leq p_i < i$).

В последних q строках содержатся запросы, каждый из которых задается парой целых чисел v_i и d_i — номером члена организации и интересующего уровня подчиненных ($1 \leq v \leq n$; $0 \leq d \leq n - 1$).

Формат выходных данных

Выведите q строк, в i -й из которых должен идти ответ на i -й запрос — целое число от 0 до n .

Пример

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

Задача К. Темпоральные дожди

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

Иногда в Пустоте идут темпоральные дожди. Это достаточно необычное событие для места «вне времени», поэтому и последствия тоже значимые — некоторые сосланные в Пустоту существа справились выбраться из нее, изучая поведение этих темпоральных дождей.

Если мы представим поверхность Пустоты как координатную решетку размером $n \times m$ (n строк и m столбцов), с уровнем поверхности в ячейке (i, j) равным $h_{i,j}$, то после темпорального дождя темпоральная энергия скапливается на этой решетке, как скапливалась бы вода при обычном дожде: если ей некуда перелиться за границы Пустоты, она будет держаться на одном уровне.

Более формально, скажем, что $d_{i,j}$ — *уровень энергии* в ячейке (i, j) — это величина со следующими свойствами:

- $d_{i,j} \geq h_{i,j}$;
- не существует пути из (i, j) до границы Пустоты по ячейкам, где каждая следующая имеет общую сторону с предыдущей, уровень поверхности каждой из которых меньше $d_{i,j}$;
- $d_{i,j}$ максимальная возможная, удовлетворяющая этому условию.

Иными словами, рассмотрим все пути от ячейки до границы Пустоты (каждый переход на пути — в соседнюю по стороне ячейку), на каждом пути выберем \max уровня поверхности, и среди всех таких выберем минимальный — это и будет уровень энергии в этой ячейке.

Определите суммарный уровень энергии относительно уровня поверхности, который установится по всей Пустоте после окончания темпорального дождя, чтобы помочь Дэдпулу и Росомахе из нее выбраться.

Формат входных данных

В первой строке даны два целых числа n и m — размеры Пустоты ($1 \leq n, m \leq 10^6$; $1 \leq n \cdot m \leq 10^6$).

В i -й из следующих n строках перечислены m чисел $h_{i,j}$ ($0 \leq h_{i,j} \leq 10^9$).

Формат выходных данных

Выведите единственное число — $\sum_{i,j} (d_{i,j} - h_{i,j})$ для удовлетворяющих условию d .

Примеры

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

Задача L. Эпилог

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

После событий фильма Уэйд и Логан наконец-то смогли спокойно отдохнуть от всего происходящего хаоса. Как вариант — для этого вполне можно устроить вечеринку, на которую позвать всех причастных (даже X-23 обещала прийти!).

Однако для хорошей вечеринки надо как минимум привести в порядок дом, в котором она будет проходить. Для этого Уэйд решил обклеить стены дома фотографиями событий фильма (их не очень сложно достать, когда ты постоянно ломаешь четвертую стену).

Сейчас у него есть n квадратных фото с длинами сторон a_1, \dots, a_n . И по ощущениям, они не помещаются на стену. Некоторые фото придется заменить, чтобы их суммарная площадь, то есть $\sum a_i^2$, была в точности равна m .

Marvel согласилась на следующие правила обмена: фото со стороной a_i можно поменять на фото со стороной $b_i < a_i$, но тогда бюджет следующего фильма уменьшится на $(a_i - b_i)^2$. Обмены можно совершать только с фото, которые были у Дэдпула изначально: например, нельзя девять раз поменять фото со стороной 10, чтобы получить фото со стороной 1: после первого же обмена полученное фото со стороной 9 будет необменным.

Определите минимальный бюджет следующего фильма, которым придется пожертвовать, чтобы получить фото суммарной площади m , или определите, что это невозможно. Все фото на руках должны быть использованы, b_i для каждого a_i Уэйд может выбрать сам.

Формат входных данных

Первая строка ввода содержит два целых числа n и m — количество фото и площадь стены, соответственно ($1 \leq n \leq 10$; $1 \leq m \leq 10\,000$).

В i -й из следующих n строк содержится число a_i — длина стороны i -го фото ($1 \leq a_i \leq 100$).

Формат выходных данных

Выведите единственное целое число — ответ на задачу, или -1 , если получить площадь m невозможно.

Примеры

стандартный ввод	стандартный вывод
3 6 3 3 1	5