Задача A. Посиделки

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дарья любит устраивать небольшие зимние посиделки с друзьями. В этом году она решила сделать встречу более уютной и принести одно из угощений, которое традиционно выбирает для таких вечеров. Однако из-за ограниченного бюджета Дарья может купить только один предмет.

У неё есть три варианта, упорядоченные по степени предпочтения:

- **Watermelon** — любимое угощение Дарьи.
- **Pomegranates** — второй по её предпочтению вариант.
- **Nuts** — третий, но всё ещё приемлемый выбор.

Дарья хочет купить ровно один из этих продуктов. Если её бюджет позволяет, она выбирает первый возможный вариант по списку. Если денег не хватает ни на один предмет, она откажется от покупки.

Формат входных данных

Ввод содержит:

- первую строку — целое число b ($0 \leqslant b \leqslant 10^6$), обозначающее бюджет Дарьи,
- следующие три строки — цены на *Watermelon*, *Pomegranates* и *Nuts* соответственно. Каждая цена представлена в виде неотрицательного целого числа, не превышающего 10^6.

Формат выходных данных

Выведите название первого предмета, который Дарья может себе позволить согласно её списку предпочтений: “Watermelon”, “Pomegranates” или “Nuts”. Если её бюджета недостаточно для покупки любого из предметов, выведите “Nothing”.

Пример

стандартный ввод	стандартный вывод
150 200 95 130	Pomegranates

Задача B. Выключатели света

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Старое исследовательское учреждение Института Нортвуда было построено в спешке, и его внутреннее устройство это отражает: это лабиринт из N камер, соединенных узкими коридорами, размещение которых не подчиняется никакой архитектурной логике. Каждая камера оборудована лампой, но лампы ненадежны и освещают только определенные коридоры, ведущие из камеры. Коридор становится доступным, если хотя бы одна из двух камер на его концах освещает его.

Доктор Лиора, последний оставшийся исследователь на месте, начинает в камере 1. Ее лампа в данный момент включена, в то время как все остальные камеры темные. Ей необходимо добраться до камеры N, где ее ждет работающий терминал, но она отказывается входить в любой коридор, который не освещен в момент, когда она пытается пройти через него. Она может свободно переключать лампу в камере, в которой она в данный момент находится, включая или отключая освещение, которое она предоставляет.

Ее конечная цель — находиться в камере N с выключенными всеми лампами, кроме лампы в камере N. Чтобы добраться туда, она может бродить по учреждению, переключая лампы по мере необходимости, при условии, что она никогда не проходит через неосвещенный коридор. Ей интересно, возможна ли такая последовательность действий вообще. Если да, она также хочет знать минимальное количество различных ламп, которые ей придется когда-либо включить во время своего путешествия.

Формат входных данных

Первая строка содержит целое число N ($1 \leqslant N \leqslant 500$), количество камер. Каждая из следующих N строк описывает схему освещения одной камеры. Строка i начинается с целого числа s_i ($0 \leqslant s_i < N$), за которым следуют s_i различных целых чисел $a_1, \ldots, a_{s_i}$ ($1 \leqslant a_j \leqslant N$, $a_j \neq i$), представляющих камеры, чьи соединяющие коридоры освещены, когда лампа в камере i включена. Пусть $M = \sum_i s_i$; оно удовлетворяет $0 \leqslant M \leqslant 2000$.

Формат выходных данных

Если добраться до камеры N с выключенными всеми другими лампами невозможно, выведите “`No`”. В противном случае выведите одно целое число: минимальное количество различных ламп, которые должны быть включены в какой-то момент по пути.

Примеры

стандартный ввод	стандартный вывод
5 2 2 3 1 4 2 4 1 1 5 1 3	3
4 1 2 2 3 4 1 2 1 3	No

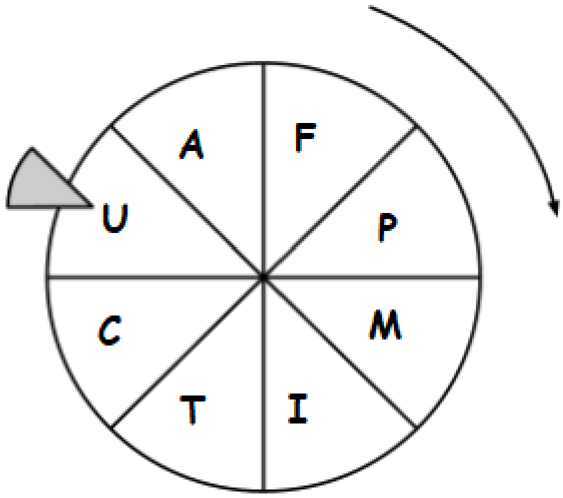
Замечание

Можно использовать камеры 3 и 4 для передачи света, и так добраться до камеры 5. Затем вернуться назад, чтобы выключить промежуточные лампы, в конечном итоге оказавшись в камере 5 с включенной только ее лампой. В итоге включены три разные лампы. Во втором примере лампу в камере 1 никогда нельзя выключить, покидая ее, поэтому целевая конфигурация не может быть достигнута.

Задача C. Колесо Букв

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Сева обнаружил старую карнавальную машину на чердаке: круговое колесо, разделенное на N секторов. Каждый сектор изначально имел уникальную заглавную английскую букву, но надписи со временем стерлись. Колесо вращается только по часовой стрелке, в то время как фиксированный указатель над ним всегда остается неподвижным.



Недавно Денис экспериментировал с колесом, прежде чем буквы полностью исчезли. Для каждого вращения он наблюдал, сколько раз буква под указателем менялась по мере вращения колеса, и записывал, какая буква в конечном итоге была отмечена указателем, когда вращение остановилось. Поскольку ни одна буква не встречалась на колесе более одного раза, каждая позиция уникально соответствовала букве в тот момент.

Теперь Сева пытается восстановить первоначальное расположение букв. Он знает общее количество секторов и имеет последовательность наблюдений Дениса, записанную в точном порядке, в котором происходили вращения. Исходя только из этой информации, он должен выяснить, какие буквы занимают какие сектора. Всякий раз, когда данные не указывают на конкретную позицию уникально, буква для этого сектора остается неизвестной.

Ваша задача — определить, существует ли хотя бы одно колесо, чьи буквы согласуются со всеми записанными вращениями. Если такое колесо не может существовать, восстановление Севы полностью проваливается. Если восстановление возможно, окончательное расположение должно быть представлено в виде цикла букв, начиная с буквы, которая появилась под указателем после последнего вращения, и продолжая по часовой стрелке вокруг колеса. Каждый сектор, буква которого не может быть однозначно определена, должна быть отмечена знаком вопроса.

Формат входных данных

Первая строка содержит два целых числа N ($2 \leqslant N \leqslant 25$) и K ($1 \leqslant K \leqslant 100$), количество секторов на колесе и количество вращений, выполненных Денисом. Каждая из следующих K строк содержит целое число S ($1 \leqslant S \leqslant 100$), количество раз, когда указатель изменил свою букву во время этого вращения, за которым следует одна заглавная буква, представляющая букву под указателем, когда колесо остановилось.

Формат выходных данных

Если ни одно колесо не соответствует всем наблюдениям, выведите "`!`". В противном случае выведите буквы колеса, начиная с буквы, указанной после последнего вращения, и двигаясь по часовой стрелке, записывая "`?`"для каждого сектора, буква которого не может быть определена.

Примеры

стандартный ввод	стандартный вывод
8 8 4 T 3 P 7 M 7 I 6 C 5 F 1 A 9 U	UAFPMITC
3 3 1 A 2 B 3 C	!

Задача D. Антикризисный депозит

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Недавно Денис запустил рекламный банковский продукт под названием *Антикризисный депозит*. Предложение быстро стало популярным среди долгосрочных клиентов, и теперь Сева, единственный программист банка, должен реализовать модуль расчета для него. Депозит следует необычной схеме начисления процентов, поэтому точность имеет решающее значение.

Депозит может быть открыт в любой день года 2025, и он должен завершиться не позднее 31 декабря того же года. Годовая процентная ставка составляет p процентов. Каждый раз, когда баланс остается неизменным в течение непрерывного периода в d дней, и баланс в начале этого периода составлял x рублей, сумма на конец периода становится $x \cdot \left(1 + \frac{p}{100} \cdot \frac{d}{365}\right)$.

Проценты добавляются в конце каждого календарного месяца или в последний день депозита, в зависимости от того, что наступит раньше. Когда проценты добавляются, они немедленно объединяются с балансом. Например, если баланс 1 мая составлял x, то 31 мая начисленные проценты $x \cdot \frac{p}{100} \cdot \frac{31}{365}$ добавляются, и 1 июня баланс становится $x \cdot \left(1 + \frac{p}{100} \cdot \frac{31}{365}\right)$, после чего проценты за июнь рассчитываются на основе этого обновленного баланса.

Если депозит заканчивается раньше, скажем, 20 мая, то проценты за ровно 20 дней добавляются в эту дату, и окончательная выплата составляет $x \cdot \left(1 + \frac{p}{100} \cdot \frac{20}{365}\right)$.

Начальный день депозита может находиться в середине месяца. Например, депозит, открытый 18 февраля, накапливает проценты в течение 11 дней до 28 февраля; если он открыт 28 февраля, то проценты за 1 день добавляются в ту же дату. Учитывая дату открытия и продолжительность депозита, Сева должен вычислить окончательную сумму, которую Денис должен выплатить клиенту, предполагая, что начальный баланс составлял x рублей.

Формат входных данных

Первая строка содержит три целых числа: начальный баланс x, годовую ставку p и продолжительность d депозита в днях ($1 \leqslant x \leqslant 100000$, $1 \leqslant p \leqslant 200$, $1 \leqslant d \leqslant 365$). Вторая строка содержит дату открытия в формате `DD-MM-YYYY`. Все даты находятся в 2025 году, и дата закрытия также принадлежит 2025 году.

Формат выходных данных

Выведите окончательный баланс депозита. Ответ будет проверяться с абсолютной или относительной точностью 10^{-6}.

Примеры

стандартный ввод	стандартный вывод
10000 15 90 01-01-2025	10374.43657773311
10000 15 30 28-11-2025	10123.424469881778
10000 15 1 30-11-2025	10004.109589041096

Задача E. Разделение дорог

Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Берляндия содержит n городов, соединенных m дорогами. Каждая дорога представляет собой прямой отрезок, соединяющий два различных города; ни два города не имеют одинаковых координат, и между любой парой городов может быть не более одной дороги. Все дороги асфальтированы, и стоимость обслуживания дороги пропорциональна ее длине. Правительство решило разделить обслуживание между двумя компаниями так, чтобы общая длина, обслуживаемая первой компанией, равнялась общей длине, обслуживаемой второй компанией. Разные части одной и той же дороги могут быть назначены разным компаниям.

Разделение будет выполнено с помощью одной прямой линии. Каждая часть дороги, которая строго лежит с одной стороны линии, назначается одной компании, а каждая часть, лежащая строго с другой стороны, назначается другой компании. Если дорога лежит точно на линии, то половина этой дороги назначается каждой компании.

Ваша задача — найти любую прямую линию, для которой общие длины частей дорог по обе стороны равны.

Формат входных данных

Первая строка содержит два целых числа n и m ($2 \leqslant n \leqslant 100000$, $1 \leqslant m \leqslant 100000$). Каждая из следующих n строк содержит по два целых числа: координаты города (x_i, y_i), при этом $|x_i|, |y_i| \leqslant 10^3$. Все координаты являются целыми числами, и все позиции городов различны. Каждая из следующих m строк содержит по два целых числа u и v ($1 \leqslant u, v \leqslant n$, $u \neq v$), что означает, что между городом u и городом v есть дорога. Любая пара городов соединена не более чем одной дорогой.

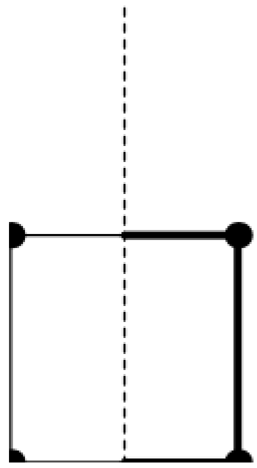
Формат выходных данных

Выведите координаты двух различных точек, которые лежат на допустимой разделяющей линии. Первая строка должна содержать координаты одной точки, вторая строка — координаты другой точки. Координаты могут быть вещественными числами, должны удовлетворять $|x|, |y| \leqslant 10^3$, и расстояние между двумя выходными точками должно быть не менее 1. Печатаемые числа могут иметь любое количество знаков после запятой; судья проверяет равенство общих длин с абсолютной ошибкой до 10^{-4}.

Пример

стандартный ввод	стандартный вывод
4 4 0 0 0 1 1 1 1 0 1 2 2 3 3 4 4 1	0.5 0.0 0.5 1.0

Замечание



В примере есть четыре города, образующие единичный квадрат, и четыре ребра, образующие цикл. Одна допустимая разделяющая линия — вертикальная линия $x = 0.5$; тогда левые и правые части дорог имеют равную общую длину. Пример вывода дает две точки на этой линии.

Задача F. Нарезка Брауни

Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Денис приготовил большой прямоугольный поднос с брауни для предстоящего кулинарного фестиваля. Поднос разделен на сетку размером $R \times C$ из маленьких квадратов брауни. Квадрат в строке i, столбце j содержит N_{ij} шоколадных чипсов.

Денису нужно распределить брауни между $A \times B$ гостями. Сначала он делает ровно $A - 1$ горизонтальных разрезов вдоль целых линий сетки, разделяя поднос на A горизонтальных полос. Затем каждая полоса нарезается независимо, делая ровно $B-1$ вертикальных разрезов вдоль целых границ. В конце брауни делятся на $A \times B$ прямоугольных кусочков.

После завершения нарезки $A \times B - 1$ гостей каждый берёт ровно один кусок. Они всегда выбирают куски жадно: гости берут куски с наибольшим количеством шоколадных чипсов, оставляя Денису тот кусок, который содержит наименьшее количество чипсов. Денис хотел бы выбрать разрезы так, чтобы этот последний кусок был как можно более богат шоколадными чипсами.

Ваша задача — определить максимальное количество шоколадных чипсов, которое Денис может гарантировать для себя, предполагая, что он выбирает разрезы оптимально.

Формат входных данных

Первая строка содержит четыре целых числа R, C, A и B ($1 \leqslant R, C \leqslant 500$, $1 \leqslant A \leqslant R$, $1 \leqslant B \leqslant C$). Каждая из следующих R строк содержит C целых числа N_{ij} ($0 \leqslant N_{ij} \leqslant 4000$), представляющих шоколадные чипсы в сетке брауни.

Формат выходных данных

Выведите одно целое число: максимальное количество шоколадных чипсов, которое Денис может гарантировать получить на своем кусочке брауни.

Пример

<table>
<tr><th>стандартный ввод</th><th>стандартный вывод</th></tr>
<tr><td>5 4 4 2
1 2 2 1
3 1 1 1
2 0 1 3
1 1 1 1
1 1 1 1</td><td>3</td></tr>
</table>

Задача G. Тур по МФТИ

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вдоль национальной дороги, называемой дорогой МФТИ, которая простирается на 10^{100} единиц с востока на запад, расположены N картин. Каждая картина содержит один символ из множества $\{\mathtt{M}, \mathtt{I}, \mathtt{P}, \mathtt{T}\}$. Андрей Михайлович, гид, хочет провести посетителей, выбрав и посетив четыре картины, символы которых, в порядке, являются 'M', 'I', 'P' и 'T'.

i-я картина находится на позиции A_i, измеренной от западного конца дороги, и содержит символ C_i. У Андрея Михайловича есть Q планов тура. В каждом плане она начинает с S_j, посещает одну картину с символом 'M', затем одну с символом 'I', затем одну с символом 'P', затем одну с символом 'T', и, наконец, отправляется в T_j. Все перемещения должны оставаться в пределах сегмента дороги.

Для каждого плана вычислите минимально возможное общее расстояние путешествия.

Формат входных данных

Первая строка содержит целое число N ($4 \leqslant N \leqslant 100{,}000$).

Каждая из следующих N строк содержит целое число A_i и символ C_i ($1 \leqslant A_i \leqslant 10^{15}$; $C_i \in \{\mathtt{M}, \mathtt{I}, \mathtt{P}, \mathtt{T}\}$; $A_1 < A_2 < \cdots < A_N$). Каждый из четырех символов появляется как минимум один раз.

Следующая строка содержит целое число Q ($1 \leqslant Q \leqslant 100{,}000$).

Каждая из следующих Q строк содержит два целых числа S_j и T_j ($1 \leqslant S_j, T_j \leqslant 10^{15}$).

Формат выходных данных

Выведите Q строк. j-я строка должна содержать минимальное общее расстояние для плана j.

Пример

стандартный ввод	стандартный вывод
7 1 M 2 I 3 T 4 P 5 I 8 T 10 M 2 3 2 7 5	7 12

Задача H. Счастливые основания

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Археологи, изучающие древнюю цивилизацию, обнаружили, что её жители использовали необычные системы счисления. Вместо того чтобы полагаться на одно основание, они свободно выбирали разные основания при записи чисел. Их записи показывают, что некоторые основания считались «священными». Основание считалось священным, если при записи числа с использованием этого основания каждая цифра в представлении появлялась на специальных табличках, украшенных только символами 4 и 7. Любое десятичное число, десятичная запись которого состоит только из этих цифр, считалось священным.

Для каждого целого числа $B \geqslant 2$ существует позиционная система счисления с основанием B, использующая цифры от 0 до $B-1$. Положительное целое число A представляется в этой системе с помощью цифр $a_n, a_{n-1}, \ldots, a_0$, так что

$$A = a_n B^n + a_{n-1} B^{n-1} + \cdots + a_0,$$

и записывается ими слева направо. Например, число 255 можно записать в основании 52 как

$$255 = 4 \cdot 52 + 47,$$

так что его представление состоит из двух цифр: 4 и 47.

Основание B считается священным для числа A, если каждая цифра в представлении A в основании B является священным десятичным числом (например, 4, 7, 44, 47, ...). Ученые хотят знать, сколько различных оснований священны для данного числа.

Вам дано целое число n. Определите, сколько оснований $B \geqslant 2$ священны для n. Если таких оснований бесконечно много, выведите -1.

Формат входных данных

Входные данные содержит одно целое число n ($1 \leqslant n \leqslant 10^{16}$).

Формат выходных данных

Выведите количество священных оснований для n, или -1, если это число бесконечно.

Пример

стандартный ввод	стандартный вывод
255	2

Замечание

В этом примере целое число 255 имеет ровно два священных основания: 52 и 62.

Задача I. Все, что вы можете выбрать

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Местное исследовательское учреждение с ограниченным временем предлагает "экспериментальный буфет". Доступно N различных экспериментальных модулей. Запуск модуля i требует A_i минут и приносит B_i единиц научного понимания.

Вы можете активировать модули по одному в соответствии со следующими правилами:

- Как только модуль активирован, он становится доступным немедленно, и вы должны запустить его до завершения.
- Вы не можете активировать один и тот же модуль более одного раза.
- Вы не можете активировать новый модуль, пока текущий не завершится.
- Точно через $T - 0.5$ минут после активации вашего первого модуля система управления блокируется: новые модули не могут быть активированы после этого момента. Однако вы можете продолжать запускать модуль, который вы в данный момент выполняете.

Ваш общий научный прирост — это сумма B_i по всем модулям, которые вы запускаете. Определите максимальный возможный прирост.

Формат входных данных

Первая строка содержит два целых числа N и T ($2 \leqslant N \leqslant 3000$, $1 \leqslant T \leqslant 3000$). Каждая из следующих N строк содержит целые числа A_i и B_i ($1 \leqslant A_i \leqslant 3000$, $1 \leqslant B_i \leqslant 3000$).

Формат выходных данных

Выведите максимальный общий научный прирост, который можно достичь.

Пример

стандартный ввод	стандартный вывод
2 60 10 10 100 100	110

Замечание

Если первый модуль запущен в момент времени 0, он завершится в момент времени 10, и второй модуль все еще может быть активирован в момент времени $10 < 59.5$, поэтому оба могут быть завершены.

Задача J. Парковочная теория

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В МФТИ есть прямоугольная парковка размером $n \times m$, состоящая из парковочных мест, расположенных в виде таблицы. Каждая строка и каждый столбец парковки имеют въезд с обеих сторон.

Парковка полностью заполнена, и для каждого места известно, в каком порядке на него заехала машина. Ячейка с числом 1 содержит автомобиль, въехавший первым, а ячейка с числом $n \cdot m$ — последним.

Денис выдвинул теорию о поведении машин на этой парковке. Он считает, что любая машина, въезжая на парковку с выбранной стороны (со стороны строки или столбца), движется прямо до своего места и никогда не меняет направление. Кроме того, машина не может проехать через клетку, где уже стоит другой автомобиль.

Денис хочет узнать, сколько существует прямоугольных подтаблиц (подпрямоугольников) парковки, которые удовлетворяют этим правилам. Подтаблица считается допустимой, если все машины внутри неё могли бы припарковаться, не нарушив описанные условия, учитывая только автомобили внутри этой подтаблицы.

Формат входных данных

Первая строка входных данных содержит два целых числа n и m ($1 \leqslant n, m \leqslant 500$) — количество строк и столбцов парковки. Каждая из следующих n строк содержит по m различных целых чисел — порядок въезда автомобилей. Гарантируется, что использованы числа от 1 до $n \cdot m$ без повторений.

Формат выходных данных

Выведите одно целое число — количество допустимых подтаблиц.

Пример

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

Задача K. Несмежные черные прямоугольники

Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Демид по-прежнему полон энтузиазма. Во время рисования в своем тетради в клетку он раскрасил некоторые квадраты в черный, а некоторые в белый цвет, и возникла новая задача.

Вам дано $N \times N$ таблица. Каждая ячейка либо черная, либо белая. Прямоугольник, сторонами которого являются границы ячеек, называется черным прямоугольником, если каждая ячейка внутри него черная и прямоугольник содержит как минимум две ячейки. Прямоугольники, которые содержат даже одну белую ячейку, не считаются черными прямоугольниками, и квадрат размером 1×1 также не считается черным прямоугольником.

Ваша задача — подсчитать, сколько неупорядоченных пар черных прямоугольников можно выбрать так, чтобы два прямоугольника не имели общих ячеек. Поскольку ответ может быть большим, выведите его по модулю 998244353.

Формат входных данных

Первая строка содержит целое число N ($2 \leqslant N \leqslant 1000$). Каждая из следующих N строк содержит строку из N символов, каждый символ — это '0' (белый) или '1' (черный).

Формат выходных данных

Выведите одно целое число: количество неупорядоченных пар несмежных черных прямоугольников, взятое по модулю 998244353.

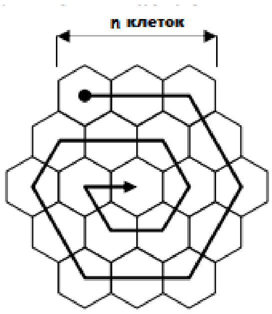
Пример

стандартный ввод	стандартный вывод
3 110 110 100	5

Задача L. Шестигранная спираль

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Большое шестигранное поле состоит из набора маленьких шестигранных клеток. Каждая сторона поля содержит n клеток. Клетки нумеруются по спирали, начиная с самой верхней ячейки, как показано на рисунке. Первые m клеток в этом порядке заполняются последовательными натуральными числами, начиная с 1, в то время как оставшиеся остаются пустыми.



Ваша задача — написать программу, которая:

- Определяет, какое число записано в q-й клетке p-й строки поля. Строки нумеруются сверху вниз, начиная с 1, а клетки в каждой строке нумеруются слева направо, начиная с 1. Если указанная клетка пуста, выведите 0.
- Определяет строку и позицию в этой строке самого большого числа в поле.

Формат входных данных

Одна строка, содержащая четыре целых числа: n, m, p, q ($1 < n \leqslant 24 \cdot 10^8$, $0 < m < 3n(n-1)$). Гарантируется, что клетка, соответствующая заданным p и q, существует.

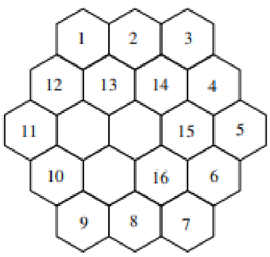
Формат выходных данных

В первой строке выведите ответ на первую задачу (число в q-й клетке p-й строки или 0, если клетка пуста). Во второй строке выведите два целых числа: номер строки и позицию клетки самого большого числа в поле.

Пример

стандартный ввод	стандартный вывод
3 16 2 3	14 4 3

Замечание



Задача M. Переключение фигур на магической доске

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вы обнаружили старую шахматную доску высотой h и шириной w, на которой некоторые клетки магически заблокированы. Демид играет на этой доске с ровно одной шахматной фигурой на доске в любой момент времени. В начале фигура — это король, расположенный в верхнем левом углу $(1, 1)$. Цель состоит в том, чтобы разместить любую фигуру в нижнем правом углу (h, w) за минимально возможное время.

Во время игры Демид может в любой момент заменить текущую фигуру на другую фигуру, которая находится на той же клетке. Замена текущей фигуры на определённую новую фигуру занимает фиксированное количество времени (секунд), в зависимости от того, какую фигуру он выбирает. Каждый фактический ход фигуры (законный шахматный ход текущей фигуры) занимает ровно 1 секунду. Замены и ходы выполняются последовательно, и их время суммируется. Фигуру никогда нельзя разместить на заблокированной клетке. Скользящие фигуры не могут проходить через заблокированные клетки, в то время как прыжок коня не зависит от заблокированных клеток, через которые он проходит.

Времена замены и правила движения следующие.

Король: замена любой фигуры на короля занимает 1 секунду. Король может перемещаться за один шаг на любую из до восьми соседних клеток (если эта клетка не заблокирована).

Слон: замена любой фигуры на слона занимает 2 секунды. Слон за один ход может пройти любое положительное количество клеток по диагонали, при условии, что ни одна из клеток на пути (включая конечную) не заблокирована.

Ладья: замена любой фигуры на ладью занимает 3 секунды. Ладья за один ход может пройти любое положительное количество клеток горизонтально или вертикально, при условии, что ни одна из клеток на пути (включая конечную) не заблокирована.

Конь: замена любой фигуры на коня занимает 4 секунды. Конь за один ход прыгает в привычной L-образной форме (две клетки в одном из четырех кардинальных направлений, затем одна клетка перпендикулярно); он может перепрыгивать через заблокированные клетки, и только конечная клетка должна быть незаблокированной.

Ферзь: замена любой фигуры на ферзя занимает 5 секунд. Ферзь за один ход может пройти любое положительное количество клеток вертикально, горизонтально или по диагонали, при условии, что ни одна из клеток на пути (включая конечную) не заблокирована.

Вычислите минимальное количество секунд, необходимых для того, чтобы разместить какую-либо фигуру на клетке (h, w). Если это невозможно, выведите -1.

Формат входных данных

Первая строка содержит два целых числа h и w ($1 \leqslant h, w \leqslant 500$). Каждая из следующих h строк содержит строку длиной w из символов "." и "#": символ "." обозначает незаблокированную клетку, а "#" обозначает заблокированную клетку. Верхняя левая клетка $(1, 1)$ и нижняя правая клетка (h, w) могут быть как заблокированными, так и незаблокированными; ходы или замены, которые поместят фигуру на заблокированную клетку, не допускаются.

Формат выходных данных

Выведите одно целое число: минимальное количество секунд, необходимых для размещения любой фигуры на клетке (h, w), или -1, если это невозможно.

Примеры

стандартный ввод	стандартный вывод
3 3 ... .#. ...	3
3 5 .#### ##.## ####.	6
10 10 ...#.###.. .....##### ...##..#.# #....#.... .#.......# ##..#..... ......#... ..##..#.#. ..#..#..#. ..#..##...	8

Замечание