

Задача 1. Генеалогические кустики

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Ходислав хочет нарисовать генеалогическое древо. Для начала он решил попрактиковаться на небольших ориентированных ациклических графах без петель и кратных рёбер, в которых входящая степень каждой вершины не больше двух.

Каждую вершину такого графа Ходислав рисует прямоугольником размера 200×100 со сторонами, параллельными осям координат, каждое ребро — отрезком прямой, соединяющим центры соответствующих прямоугольников. Ориентация рёбер на рисунок не наносится. Ходислав хочет, чтобы весь рисунок помещался в прямоугольник размером 2000×1000 , при этом укладка графа была бы *изящной*:

- вершины прямоугольников должны иметь целочисленные координаты;
- прямоугольники не должны иметь общих точек, в том числе недопустимо касание вершинами прямоугольников;
- отрезки-рёбра имеют общие точки, если и только если соответствующие рёбра графа имеют общую вершину;
- отрезки-рёбра не должны иметь общих точек с прямоугольниками, кроме соответствующих инцидентным вершинам;
- все геометрические объекты должны находиться внутри или на границе прямоугольника с координатами углов $(0; 0)$, $(2000; 0)$, $(0; 1000)$, $(2000; 1000)$.

Ходиславу больше нравится рисовать, чем укладывать графы, поэтому он просит вас сделать *изящную* укладку.

Формат входных данных

В первой строке входного файла находится единственное целое число N — количество вершин графа ($1 \leq N \leq 5$). В следующих N строках описаны входящие рёбра для вершин с номерами от 1 до N : сначала указано количество входящих рёбер $k \in \{0, 1, 2\}$, далее следуют k номеров вершин, из которых выходят рёбра в текущую.

Формат выходных данных

В выходной файл в отдельной строке для каждой вершины в порядке номеров от 1 до N выведите координаты левого нижнего угла (с минимальными координатами) соответствующего прямоугольника из *изящной* укладки.

Пример

<code>input.txt</code>	<code>output.txt</code>
1 0	948 131
3 0 2 1 3 0	0 0 0 900 1800 0

Задача 2. Многомерные пути

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Далёкая-далёкая галактика представляет собой D -мерный параллелепипед, заданный набором неравенств ($0 \leq d_i \leq a_i$), где d_i — i -я координата (для i от 1 до D).

В точке, все координаты которой равны нулю, расположен корабль «Тысячелетний сокол». В некоторых целочисленных точках параллелепипеда находятся базы Империи, посещать эти точки корабль не может. За один световой час корабль может перейти по линии, параллельной одной из осей координат, в любую целую точку, не содержащую имперской базы, ровно одна координата которой больше текущей координаты на 1, а остальные координаты совпадают.

В точке, i -я координата которой равна a_i для всех i от 1 до D , находится цель, до которой планирует добраться корабль.

По заданному количеству размерностей, максимальным значениям координат и набору баз Империи определите, сколько существует различных путей, следуя которыми, «Тысячелетний сокол» сможет добраться до цели. Так как ответ может быть очень большим, выведите остаток от его деления на 998 244 353.

Формат входных данных

Первая строка входных данных содержит два целых числа N и D ($1 \leq N \leq 1000$, $2 \leq D \leq 20$) — количество имперских баз и количество измерений пространства, соответственно. Вторая строка входных данных содержит D целых положительных чисел a_i — максимальные значения каждой координаты. Гарантируется, что общее количество целых точек в галактике не превосходит 2^{24} . j -я из последующих N строк содержит по D целых чисел $b_{j,i}$, где $0 \leq b_{j,i} \leq a_i$ — i -я координата j -й базы. Гарантируется, что никакие две базы не находятся в одной точке и в начальной точке «Тысячелетнего сокола» нет имперской базы.

Формат выходных данных

Выведите одно число — остаток от деления количества различных путей на 998 244 353.

Пример

input.txt	output.txt
4 3 3 4 5 0 0 1 0 3 0 0 2 0 2 4 4	10710

Задача 3. Многоугольник

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Слесарь Полесов очень хочет построить выпуклый многоугольник с наперед заданными углами, причем не абы какой, а обязательно тот, у которого все стороны имеют длину не менее 1 и при этом наибольшая по длине сторона имеет наименьшую возможную длину. Астролябии у него нет, да и с ней он тоже с подобной проблемой не справится. Придется это сделать вам.

Формат входных данных

В первой строке входного файла записано одно целое число M — количество тестов ($1 \leq M \leq 1000$). Далее идут M блоков.

В каждом блоке в первой строке записано целое число N — количество углов многоугольника ($3 \leq N \leq 1000$).

Во второй строке N целых чисел x_i — углы в градусах искомого выпуклого многоугольника, перечисленные в порядке обхода его вершин против часовой стрелки ($1 \leq x_i \leq 179$). При этом угол x_1 при первой вершине, угол x_2 — при второй вершине и т.д.

Формат выходных данных

Для каждого теста, если ответ существует, то в первую строку нужно вывести слово **Yes**, а далее в N строках вывести по два вещественных числа — координаты вершин, перечисленные в том же порядке, что и прилегающие к ним углы, заданные во входном файле. При этом координаты вершин не должны превосходить по модулю 10^6 , а длины всех сторон должны быть не менее 1 и не более 10^6 .

Если ответа, удовлетворяющего заданным ограничениям, не существует, то необходимо вывести слово **No**.

Ответ **Yes** будет зачтен, если он удовлетворяет следующим ограничениям по точности.

- Все углы предъявленного многоугольника отличаются от заданных в условии не более чем на 10^{-4} градусов.
- Длины сторон многоугольника лежат в отрезке $[1 - 10^{-6}, 10^6 + 10^{-6}]$.
- $(a - b)/a \leq 10^{-3}$, где a — максимальная длина стороны в многоугольнике участника, а b — максимальная длина стороны в решении жюри.

Пример

input.txt	output.txt
2	Yes
4	0 1
90 90 90 90	0 0
3	1 0
70 60 100	1 1
	No

Пояснение к примеру

Если не принимать во внимание требование минимизации самой длинной стороны, то ответ

0 1

0 0

2 0

2 1

тоже подошел бы для первого подтеста. Но среди всех возможных прямоугольников с длинами всех сторон не менее 1 именно для квадрата достигается минимально возможная длина наибольшей стороны.

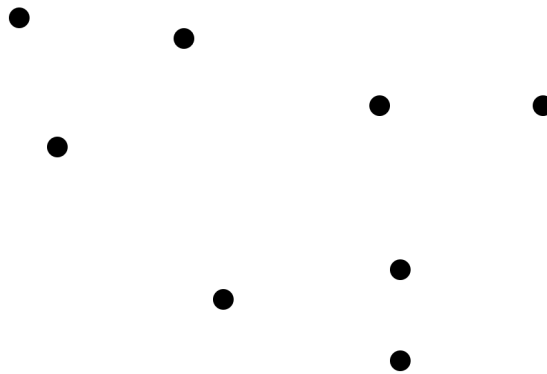
Задача 4. Вырезание

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

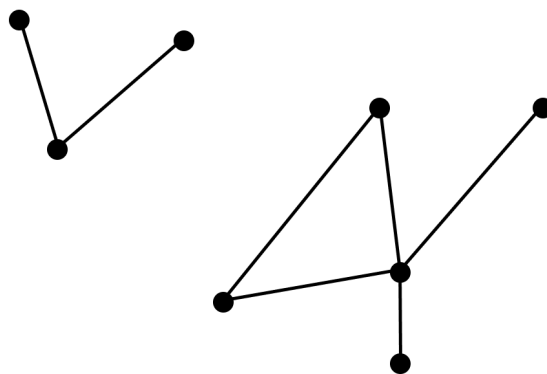
— Не порежься, — сказала мама.

— Всё будет хорошо, — ответила Катя, схватила канцелярский нож с листком бумаги и побежала в свою комнату.

На бесконечном листке бумаги Катя выбрала квадрат размером $10^9 \times 10^9$ и нарисовала в нём N случайных равномерно распределённых точек.

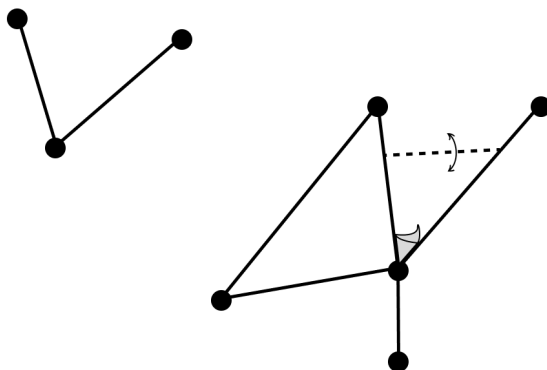


Листок Катя положила на стол (бесконечный, конечно же). Канцелярским ножом Катя проводит прямолинейные разрезы от одной нарисованной точки до другой.



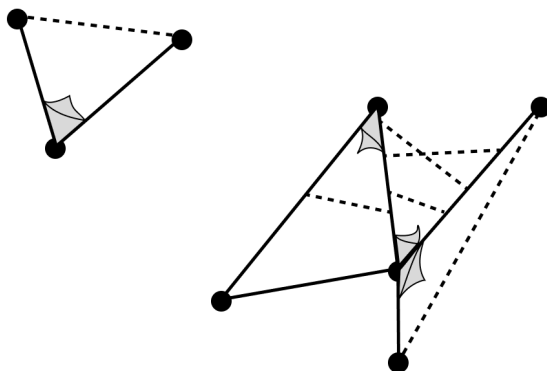
Катя может выбирать любые пары точек, и выбор этот очень важен. Чтобы мама не волновалась, Катя хочет минимизировать суммарную длину разрезов канцелярским ножом.

После того, как все разрезы ножом будут сделаны, Катя будет отгибать кусочки бумаги, которые можно отогнуть, и возвращать их на своё место. Вдоль линии загиба останется след.

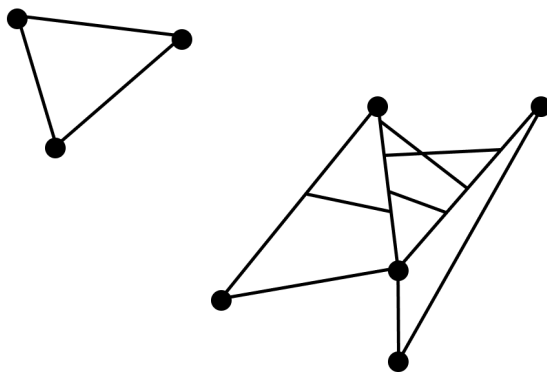


Более формально, загиб можно оставить на отрезке между точками A и B , если существует непрерывный контур из разрезов, проходящий от точки A до точки B . В таком случае этот контур и будет границей отгибаемого кусочка бумаги.

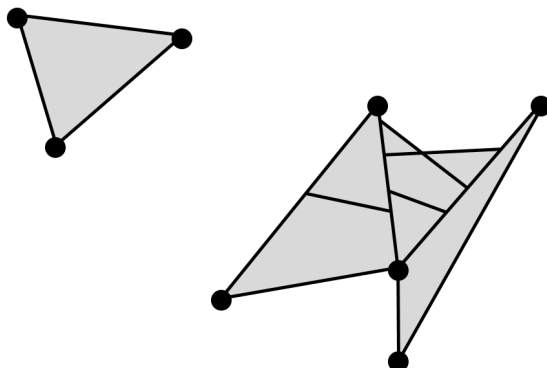
Катя оставит столько следов от загибов, сколько нужно.



Мама научила Катю аккуратно рвать бумагу вдоль линии, где бумагу когда-то загибали. Для этого надо придавить ладонями бумагу по обе стороны от линии загиба и аккуратно потянуть. Когда Катя закончит гнуть бумагу, она превратит все загибы в ровные разрывы.



Теперь Катя вынет все полностью вырезанные кусочки бумаги, и увидит серую поверхность стола.



Чего же Катя хочет добиться? Задача Кати — получить своими стараниями как можно большую по площади дырку в бумаге, и только вторым приоритетом заботиться о безопасности. Помогите Кате максимизировать площадь вырезанных кусков бумаги. Среди всех оптимальных по площади решений нужно найти такое, которое минимизирует суммарную длину разрезов **канцелярским ножом**.

Формат входных данных

В первой строке входного файла записано целое число N — число точек ($1 \leq N \leq 500$). В i -й из следующих N строк даны целочисленные координаты i -й точки, разделённые пробелом: X_i и Y_i ($0 \leq X_i, Y_i \leq 10^9$). Точки взяты случайно, независимо и равномерно из множества допустимых точек (в частности, точки могут повторяться). Набор тестов сгенерирован заранее, является одним и тем же для всех посылок и состоит не более, чем из 100 тестов.

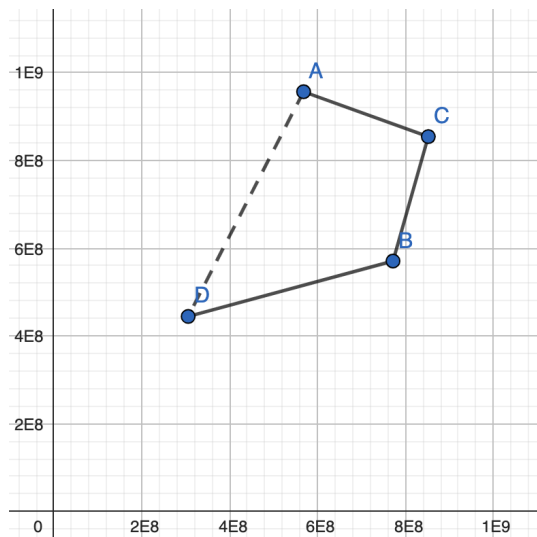
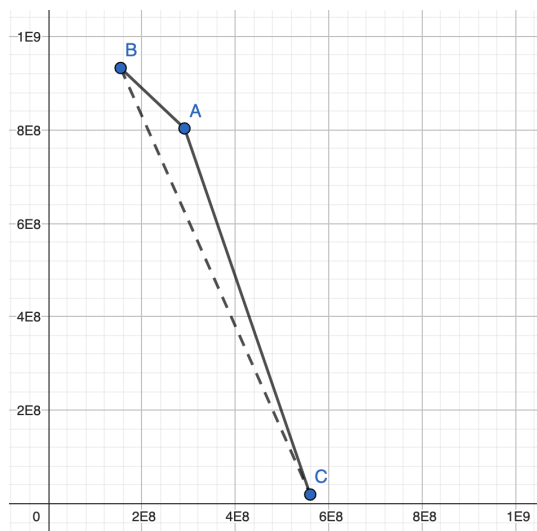
Формат выходных данных

Выведите минимальную сумму длин разрезов. Ваше решение будет зачтено, если оно отличается от решения жюри не более чем на 10^{-6} по абсолютной или относительной погрешности.

Пример

input.txt	output.txt
3 291104340 802991395 154241969 932309742 560248325 19648104	1016584295.515070796
4 568087686 956804905 771594656 570799365 851845381 854428925 305543128 444493225	1079288619.286064148

Пояснение к примеру



Иллюстрации к первому и второму примеру с отмеченными линиями оптимальных разрезов и загибов.

Задача 5. Жук

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	32 мегабайта

Герой этой задачи — всего лишь маленький жук. Жук маленький, а живёт в огромной прямоугольной комнате длины L . При этом комната разбита на N полос паркета шириной равной одной единице длины, каждая из которых идёт вдоль длины комнаты (см. рисунок). Каждая полоса в свою очередь разбита на прямоугольные элементы паркета. Вдоль i -й полосы все элементы паркета имеют длину M_i , кроме, возможно, самого первого элемента и самого последнего. Первый элемент имеет длину R_i , ну а длину последнего элемента вы и сами можете вычислить.

Между любой парой соседствующих элементов паркета есть ложбинка, через которую жуку тяжело перебираться. Поэтому во время своих путешествий жук хотел бы перебираться с одного элемента паркета на другой как можно меньшее число раз. В любой момент жук может перебираться только на такие элементы паркета, которые имеют общую границу ненулевой длины с элементом паркета, на котором жук в этот момент сидит. Перебираться между элементами паркета через углы, которыми они стыкуются, у жука не получается — там ложбинки немного шире. Посчитайте, какое наименьшее число раз жуку придётся перебраться с одного элемента паркета на другой, чтобы попасть из угла комнаты А в противоположный угол комнаты В.

Формат входных данных

В первой строке входного файла записаны через пробел два целых числа: число полос паркета N и длина комнаты L ($1 \leq N \leq 10^5$, $1 \leq L \leq 10^8$). В i -й из следующих N строк записаны через пробел два целых числа, описывающих i -ю полосу паркета: M_i и R_i ($1 \leq M_i \leq L$, $1 \leq R_i \leq M_i$). Все M_i попарно различны.

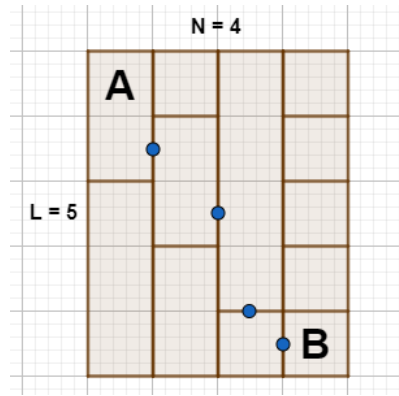
Формат выходных данных

Выведите наименьшее число переходов с одного элемента паркета на другой, которое придётся произвести жуку.

Примеры

input.txt	output.txt
4 5 3 2 2 1 4 4 1 1	4
5 10 1 1 3 2 2 2 6 2 4 1	6

Пояснение к примеру



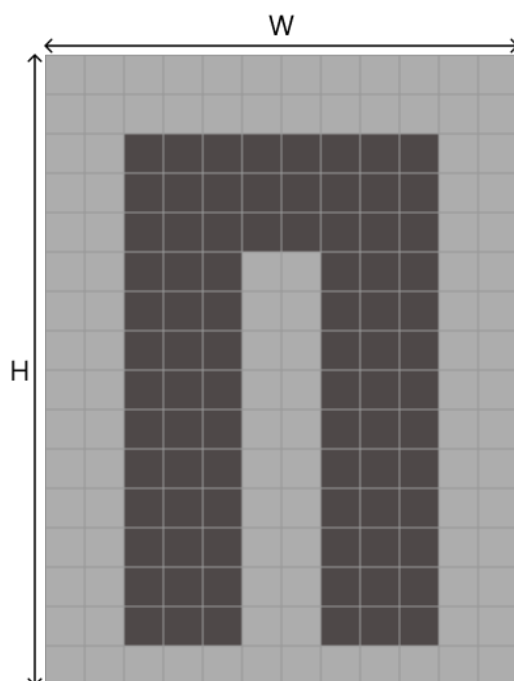
Иллюстрация, показывающая комнату из первого примера. Точками отмечены места, в которых жук перебирался между элементами паркета.

Задача 6. Узор

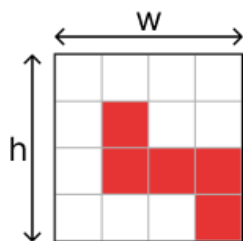
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В вышивальную мастерскую поступил новый заказ на элегантные штаны. Первым делом для вышивания элегантных штанов взяли прямоугольный кусок ткани длины H миллиметров и ширины W миллиметров. Эту ткань назовём заготовкой.

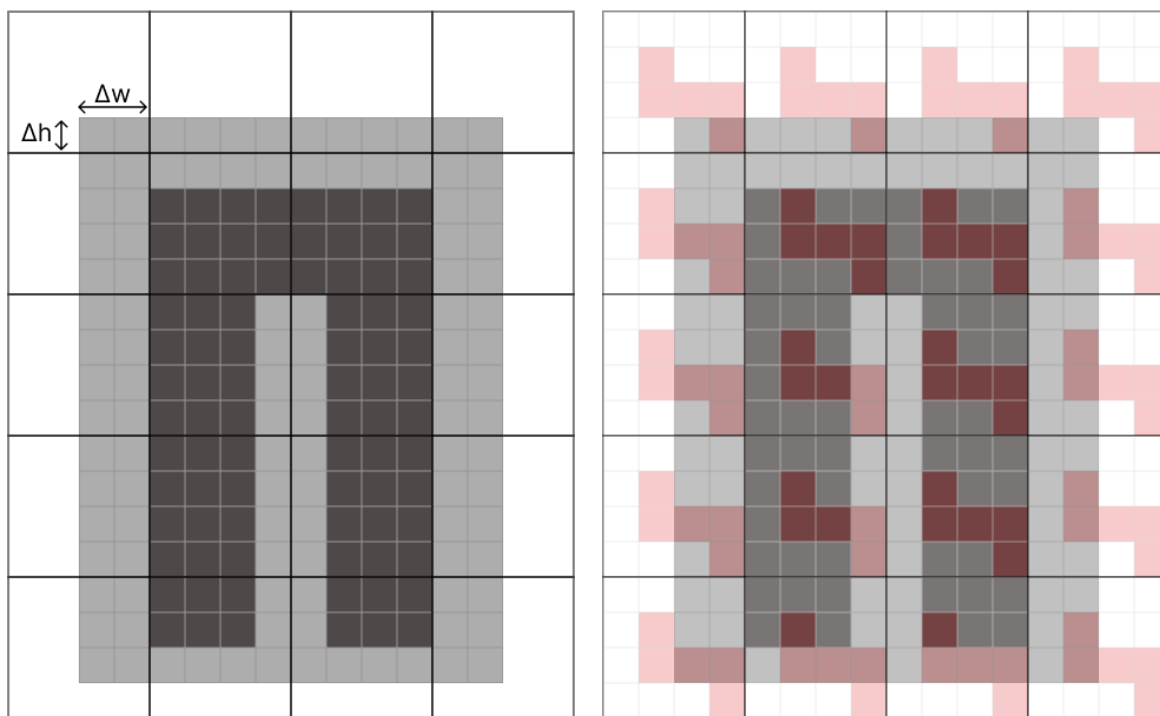
Вторым делом для вышивания элегантных штанов заготовку разбивают на квадратные клетки со стороной 1 миллиметр, и для каждого квадрата отмечают, будет он частью штанов, или попадёт в обрезки. Это называется выкройкой. Выкройку можно задать таблицей нулей и единиц, в которой единицы соответствуют квадратам ткани, попавшим в предмет одежды, а нули — попавшим в обрезки.



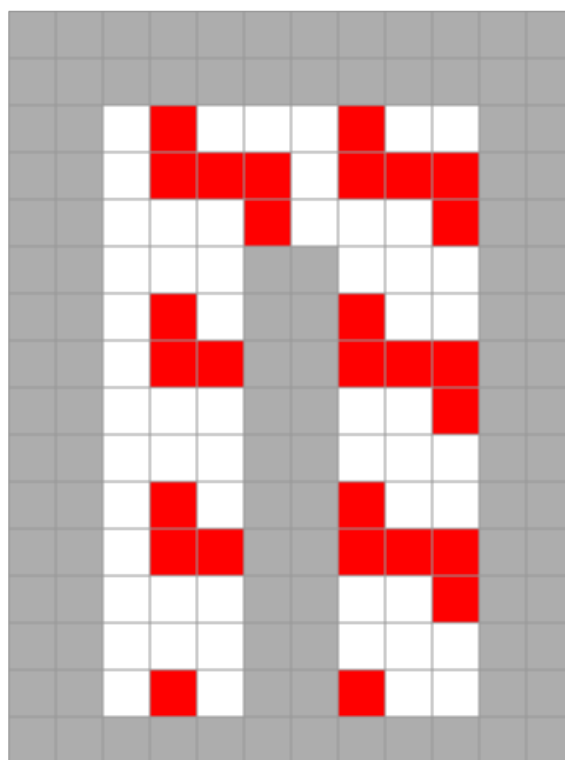
Третьим делом для вышивания элегантных штанов фиксируют шаблон вышивки. Это рисунок длины h миллиметров и ширины w миллиметров, которым замощают всю заготовку. Рисунок красно-белый, и задан таблицей нулей и единиц. Единицы соответствуют красным клеткам рисунка, а нули — белым.



Для замощения выбирается некоторое смещение шаблона относительно заготовки: Δh вдоль длины и Δw вдоль ширины.



Вот так выглядит вышивка на заготовке.



Нам не так уж важно, что делается четвёртым и пятым делом для вышивания элегантных штанов. Критическая точка процесса — выбор смещения шаблона относительно заготовки. Дело в том, что выбирая разные смещения Δh и Δw , можно получать различное число красных и белых клеток на самих штанах. Само собой элегантность штанов прямо зависит от абсолютного значения разности белых и красных клеток на штанах. Минимизируйте эту

разницу, сделав число белых клеток как можно ближе к числу красных клеток, и выдайте её в ответ.

Формат входных данных

В первой строке входного файла записаны целые числа H , W , h и w ($1 \leq H, W \leq 2000$, $1 \leq h, w \leq 100$).

В следующих H строках дано описание выкройки в виде таблицы нулей и единиц. Каждая строка таблицы имеет длину W .

Затем в h строках длины w — описание узора в виде таблицы нулей и единиц.

Формат выходных данных

Выведите минимальный модуль разницы между количеством красных и белых клеток на штанах.

Пример

input.txt	output.txt
16 12 4 4 000000000000 000000000000 001111111100 001111111100 001111111100 001110011100 001110011100 001110011100 001110011100 001110011100 001110011100 001110011100 001110011100 001110011100 001110011100 001110011100 000000000000 0000 0100 0111 0001	22

Задача 7. Очень странный сон

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Клавдии Ивановне приснился очень странный сон. Длинная-предлинная цепочка покойников укладывается по одному в гробы. Гробы все одинаковой вместимости, в каждый может войти от a до b усопших ($a < b$). Сон этот так ее взволновал, что она тут же поведала об этом Исполиту Матвеевичу.

У того тут же возникла мысль — сможет ли он всех сегодняшних старгородских покойников, так любовно им зарегистрированных, разложить аккуратно по гробам, естественно, строго соблюдая требование любимой тещи, чтобы в каждом было не более b и не менее a покойников.

Немного поразмышляв, он обнаружил, что, начиная с некоторого N_0 , мы гарантированно можем разложить любое количество $N \geq N_0$ покойников по гробам. Но вот найти это самое искомое N_0 он никак не может. Остап Ибрагимович еще на подходе, так что помогать ему некому. Вся надежда на вас.

Формат входных данных

В первой строке входного файла записано одно целое число M — количество тестов ($1 \leq M \leq 10^5$).

Далее идут M строк, в каждой из которых записано по два натуральных числа — границы вместимости гроба ($1 \leq a < b \leq 10^9$).

Формат выходных данных

В выходной файл вывести M строк, в каждой из которых должно быть записано одно целое число N_0 , такое что N_0 покойников и любое большее количество всегда можно разложить по гробам. Следует вывести **минимальное** подходящее число $N_0 \geq 1$.

Пример

<code>input.txt</code>	<code>output.txt</code>
2	3
3 5	1
1 10	

Задача 8. Похожие числительные

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Порядковые числительные в русском языке записываются следующим образом:

Сначала определим правила записи чисел от 1 до 999. Одним словом записываются следующие числа:

- Числа от 1 до 19 записываются так: один, два, три, четыре, пять, шесть, семь, восемь, девять, десять, одиннадцать, двенадцать, тринадцать, четырнадцать, пятнадцать, шестнадцать, семнадцать, восемнадцать, девятнадцать.
- Кратные 10 числа от 20 до 90 записываются так: двадцать, тридцать, сорок, пятьдесят, шестьдесят, семьдесят, восемьдесят, девяносто.
- Кратные 100 числа от 100 до 900 записываются так: сто, двести, триста, четыреста, пятьсот, шестьсот, семьсот, восемьсот, девятьсот.

Если число больше 100, то сначала записывается количество полных сотен в числе. Если число не делится на 100 нацело, затем через пробел записывается число, равное остатку от деления числа на 100.

Если число больше или равно 20, то сначала записывается количество полных десятков в числе, если число не делится на 10 нацело, то через записывается остаток от деления на 10.

Если число меньше 20, оно записывается одним словом.

Например, 239 записывается как «двести тридцать девять» (две полных сотни — записываем «двести», затем записываем остаток от деления на 100, равный 39, то есть «тридцать» и «девять» как остаток от деления на 10), 105 как «сто пять» (одна полная сотня — записываем «сто», остаток равен пяти и тем самым записывается в одно слово).

Числа, большие 999, записываются по следующему правилу: разбиваем десятичную запись числа на блоки по три цифры справа налево. Если все цифры — нули, соответствующий блок является пустым. Иначе записываем получившееся число строкой S (игнорируя ведущие нули) по предыдущему правилу, далее, в зависимости от номера блока:

- Первый блок — ничего не делаем;
- Второй блок — если S заканчивается словами «один» или «два», они заменяются на «одна» или «две». Далее к S через пробел дописывается слово «тысяча», если S заканчивается словом «одна», «тысячи», если S заканчивается словами «две», «три» или «четыре», и «тысяч» во всех остальных случаях.
- Третий блок — к S через пробел дописывается слово «миллион», если S заканчивается словом «один», «миллиона», если S заканчивается словами «два», «три» или «четыре», и «миллионов» во всех остальных случаях.
- Четвёртый блок — к S через пробел дописывается слово «миллиард», если S заканчивается словом «один», «миллиарда», если S заканчивается словами «два», «три» или «четыре», и «миллиардов» во всех остальных случаях.
- Пятый блок — к S через пробел дописывается слово «триллион», если S заканчивается словом «один», «триллиона», если S заканчивается словами «два», «три» или «четыре», и «триллионов» во всех остальных случаях.

После чего записываем блоки в том же порядке, в котором они шли в десятичной записи. При этом любые два соседние слова разделены ровно одним пробелом.

Например, число 1 000 345 654 321 записывается как «один триллион триста сорок пять миллионов шестьсот пятьдесят четыре тысячи триста двадцать один»: первый блок — «триста двадцать один», во втором $S = 654$, заканчивается на «четыре» — дописываем слово «тысячи», в третьем $S = 345$, заканчивается на «пять», дописываем «миллионов», четвёртый блок состоит из всех нулей и поэтому пуст, в пятом $S = 1$, дописываем «триллион» и записываем блоки от пятого к первому слева направо.

Число 2023 записывается как «две тысячи двадцать три»: первый блок — «двадцать три», во втором $S = 2$, заменяем слово «два» на «две» и дописываем слово «тысячи».

Вам даны два целых положительных числа L и R , $L < R$, строго меньших 10^{15} . Ваша задача — найти пару таких чисел a и b между L и R включительно, что длина d наибольшей общей подпоследовательности соответствующих порядковых числительных с учётом пробелов в русском языке является **максимальной** для всех таких пар на заданном отрезке, и вывести полученное значение d .

Напомним, что общая подпоследовательность двух строк — это строка, которую можно получить из каждого из этих слов удалением некоторого (возможно, пустого) набора символов. Например, для строк «двести три» и «девяносто пять» длина наибольшей общей подпоследовательности равна 6: подойдёт любая из 6-символьных строк «двст т» и «дест т».

Формат входных данных

Первая строка входных данных содержит два целых числа L и R ($1 \leq L < R < 10^{15}$).

Формат выходных данных

Выведите одно целое число — ответ к задаче.

Примеры

input.txt	output.txt
1 4	2
271828 314159	59

Задача 9. Е-бюрократ

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

Это интерактивная задача

Вы реализуете для корпорации «Геркулес» систему «Е-бюрократ», которая принимает на вход заявки от различных отделов, ставит случайную резолюцию, подписывает их электронной подписью директора корпорации и возвращает назад.

К сожалению, разные отделы не придерживаются каких-то общих стандартов в наименовании заявок, хотя заявки от одного учреждения имеют один и тот же формат и в рамках этого формата всегда нумеруются последовательными целыми числами, начинающимися с единицы.

Более формально, формат названия заявки в каждом отделе — это строка длины не более, чем 80 символов, состоящая из строчных букв, цифр и **ровно одной** *форматной группы*. Форматная группа имеет или вид `%d`, или вид `%0xd`, где x — цифра от 2 до 9. В первом случае номер заявки выводится вообще без ведущих нулей, во втором случае, если номер заявки имеет длину менее x знаков, то выводятся ведущие нули так, чтобы длина номера была ровно x , в противном случае ведущие нули не выводятся. Например, если формат названия — `berlaga1bukh%02d`, то заявка 1 будет называться `berlaga1bukh01`, заявка 42 — `berlaga1bukh42`, заявка 322 — `berlaga1bukh322`.

Заметим, что форматная группа может быть как в начале названия файла, так и в конце, а также следовать после цифр (например, форматы `%dzayavka`, `zayavka%09d`, `agent007%02drequest` являются корректными).

Вы хотите узнать формат для конкретного отдела. Для этого вы можете запросить, как будет называться заявка с номером x , где x — произвольное целое число от 1 до $10^9 - 1$. Чтобы не отвлекать сотрудников от важной работы по написанию заявок, вы можете сделать не более двух запросов.

Протокол взаимодействия

Взаимодействие начинает Ваша программа, задавая запрос в формате `? id`, где id — целое число, задающее номер заявки ($1 \leq id < 10^9$). Программа жюри в ответ выдаст строку, в которой произведено форматное преобразование. Если вы хотите вывести ответ, выведите `! s`, где s — формат названия для данного отдела. Это действие запросом не считается.

Заметим, что в данной задаче ответ программы жюри может формироваться по мере поступления запросов (но при этом он всегда будет соответствовать всем ранее сделанным запросам), то есть интерактор **является адаптивным**.

Пример

Стандартный ввод	Стандартный вывод
<code>roga3333kopyta</code>	<code>? 333</code>
<code>roga322kopyta</code>	<code>? 22</code>
	<code>! roga3%02dkopyta</code>

Задача 10. НГУголок

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам дана прямоугольная таблица $R \times C$. В каждой клетке этой таблицы записывается одна из букв Н, Г и У. Вычислите вероятность того, что в таблице встретится **хотя бы одна** комбинация

НГ
*У

где * — любая буква.

Формат входных данных

Первая строка входных данных содержит два целых числа R и C — количество столбцов и строк в таблице, соответственно ($1 \leq R, C \leq 128$, $1 \leq \min(R, C) \leq 14$).

Формат выходных данных

Можно показать, что вероятность является рациональным числом. Запишем его в виде несократимой дроби p/q , где $p \geq 0$, $q > 0$.

Выведите остаток от деления числа $p \cdot q^{-1}$ на 998 244 353 (здесь q^{-1} — число, которое при умножении на q даёт в остатке 1 при делении на 998 244 353).

Примеры

<code>input.txt</code>	<code>output.txt</code>
2 1	0
2 4	241002752

Задача 11. Угадай фигуру

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

Вы играете с компьютером в следующую игру на стандартной шахматной доске 8×8 . Изначально доска пуста. Обозначения полей на этой доске соответствуют стандартной шахматной нотации: поля обозначаются парой «буква-цифра», например, левое нижнее поле обозначается `a1`.

```
8 . . . . . . . .
7 . . . . . . . .
6 . . . . . . . .
5 . . . . . . . .
4 . . . . . . . .
3 . . . . . . . .
2 . . . . . . . .
1 . . . . . . . .
 a b c d e f g h
```

Компьютер выбирает одну из белых шахматных фигур (в этой задаче пешка также считается фигурой), какую именно — вы не знаете.

Вы выбираете любое из полей, кроме полей на верхней и нижней горизонтали, компьютер ставит фигуру на это поле и сообщает список букв, соответствующих полям, на которые фигура может пойти. Вы можете или сразу назвать фигуру (если угадываете — выигрываете), или попросить поставить ещё одну неизвестную фигуру на другое поле (фигура, выбранная во второй раз, может совпадать с первоначальной или отличаться от неё; в частности, возможен вариант, что обе фигуры будут ферзями или королями) и сообщить список букв, соответствующих полям, на которые может пойти первая и вторая фигура после того, как вторая фигура была установлена. Если вы запросили вторую фигуру, вы обязаны угадать обе фигуры.

Напоминаем, что белая пешка может ходить только на одну позицию вперёд (то есть, в сторону увеличения номера горизонтали, оставаясь на той же вертикали), ладья — на любое число позиций по горизонтали или вертикали, слон — на любое число позиций по диагонали, конь — буквой «Г»: на два поля в одном направлении, после чего — на одно в перпендикулярном, король — на любое из 8 полей, имеющих с текущим общую точку, ферзь — на любое число позиций по горизонтали или диагонали.

Протокол взаимодействия

Взаимодействие начинается ваша программа, выводя знак вопроса `?`, и после него через пробел координаты поля в стандартной шахматной нотации — строку, в которой первым символом является буква от `'a'` до `'h'`, а вторым — цифра от 2 до 7 (использовать первую и восьмую горизонталь запрещается, чтобы исключить попадание на них пешки).

Компьютер выводит строку, состоящую не более, чем из восьми букв от `'a'` до `'h'`, отсортированных по возрастанию — номера вертикалей, на которые может пойти фигура. Если фигура не может сделать ход, выводится один символ `'-'`.

После этого ваша программа или выводит ответ в виде `! f1`, где `f1` — `p` для пешки, `n` для коня, `b` для слона, `r` для ладьи, `k` для короля и `q` для ферзя, или делает второй запрос в аналогичном формате.

На второй запрос компьютер выводит две строки — номера вертикалей, на которые может пойти первая фигура (с учётом возможных помех от второй), и номера вертикалей, на которые может пойти вторая фигура (с учётом возможных помех от первой).

После этого ваша программа должна вывести ответ в виде ! f_1 f_2 , где f_1 и f_2 — обозначения для первой и второй фигуры, соответственно.

Гарантируется, что компьютер не заменяет уже поставленную на доску фигуру другой, даже если ответы для них будут одинаковы (то есть, что интерактор не является адаптивным).

Пример

стандартный ввод	стандартный вывод
? e5	def
? b2	def abcdefgh
! k q	