

Problem A. Угадать число

Input file: *standard input*
Output file: *standard output*
Time limit: 6 seconds (12 seconds for Java)
Memory limit: 256 mebibytes

Это интерактивная задача.

Петя и Вася играют в игру. Игра длится несколько раундов. Каждый раунд начинается с того, что Петя загадывает число от 1 до n , где n не зависит от номера раунда. Затем столько раз, сколько он захочет, Вася выбирает число от 1 до n и спрашивает, верно ли, что Петино число меньше выбранного, а Петя отвечает ему «да» или «нет». Затем Вася называет число, которое по его мнению загадал Петя. Цель Васи — правильно угадать в каждом раунде число, загаданное Петей, а также задать суммарно как можно меньше вопросов.

Помогите Васе написать программу, которая бы играла вместо него.

Input

В первой строке даны числа n и k — максимальное число, которое может загадать Петя, и число раундов ($1 \leq n \leq 10^9$, $k = 10^4$).

Далее на каждый поступивший вопрос следует ответ Пети. Ответ на каждый вопрос — одна строка: «<», если число Пети меньше, или «>» в противном случае.

На попытку отгадать число никакого ответа не следует.

Output

Обработайте все k раундов. В каждой строке должен содержаться ровно один запрос вида

$s \ x$

s — один символ, означающий вид запроса: '?' означает вопрос, а '=' означает попытку угадать;

x — число от 1 до n .

Суммарное число запросов первого типа не должно превосходить $k(\log_2(n+1) + 0.1)$.

Examples

standard input	standard output
4 2	? 3
<	? 2
>=	= 2
	? 3
>=	? 4
>=	= 4

Note

Обратите внимание, что тест из условия некорректен, так как $k \neq 10^4$. Вместо этого первый тест будет такой: 4 10000

Problem D. Подкубы

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Вася завёл массив длины 2^n , индексированный строками длины n , состоящими из нулей и единиц. Изначально он заполнил его нулями. Затем m раз он брал какой-то шаблон, то есть строку длины n , состоящую из нулей, единиц и знаков вопроса, и присваивал всем элементам массива, индексы которых удовлетворяли данному шаблону, одно и то же число. Теперь он хочет знать сумму всех элементов массива. Помогите ему её вычислить.

Input

В первой строке даны два целых числа: n и m ($1 \leq n, m \leq 32$).

В каждой из следующих m строк через пробел записаны строка s и целое число q — шаблон очередной операции и число, которому Вася присваивал элементы массива ($0 \leq q \leq 10^9$).

Output

Выведите ответ на задачу.

Examples

standard input	standard output
3 2 ?0? 2 1?? 3	16

Problem F. Дождь

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Идёт дождь. Сферический человек хочет сесть на автобус, при этом намочиться как можно меньше. Чтобы сесть на автобус, человеку нужно оказаться в одной точке с автобусом. Вычислите наименьший объём воды, который может попасть на него. Все координаты и скорости измеряются в удавах, а размеры человека и автобуса — в мартышках. Одна мартышка пренебрежительно мала по сравнению с одним удавом, поэтому взаимодействием дождя, человека и автобуса в начале и при посадке можно пренебречь.

Input

Мульти тест. Число тестов не превосходит 10^5 . Каждый тест находится в отдельной строке и состоит из 12 целых чисел:

- c ($0 \leq c \leq 1000$) — объём воды в литрах, падающий каждый час на каждую квадратную мартышку.
- rx, ry, rz ($-1000 \leq rx, ry \leq 1000, -1000 \leq rz \leq -1$) — 3-х мерный вектор скорости дождя в удавах в час.
- R ($0 \leq R \leq 1000$) — радиус тела человека в мартышках.
- hx, hy ($-1000 \leq hx, hy \leq 1000$) — координаты изначального положения человека в удавах.
- s ($0 \leq s \leq 1000$) — максимальная скорость человека в удавах в час.
- bx, by ($-1000 \leq bx, by \leq 1000$) — координаты изначального положения автобуса в удавах.
- vx, vy ($-1000 \leq vx, vy \leq 1000$) — вектор скорости автобуса в удавах в час.

Output

Выведите ответ на каждый тест в отдельной строке. Если догнать автобус невозможно, выведите -1 , иначе выведите минимальный объём в литрах с точностью не менее 10^{-9} .

Example

standard input	standard output
100 10 0 -30 10 0 0 10 300 0 -20 0	314159.2653589793
0 10 20 -30 15 0 0 10 10 10 0 10	-1
0 10 20 -30 15 0 0 10 10 -10 0 10	0.0000000000

Problem G. Точки и круги

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

На плоскости даны n точек. Два круга называются эквивалентными, если множества точек среди этих n , которые они содержат (внутри или на границе), совпадают. Сколько всего существует неэквивалентных кругов?

Input

В первой строке дано число n ($0 \leq n \leq 500$).

В каждой из следующих n строк записано по два целых числа: x_i и y_i — координаты i -ой точки ($-30000 \leq x_i, y_i \leq 30000$).

Гарантируется, что все точки различны, никакие три точки не лежат на одной прямой и никакие четыре точки не лежат на одной окружности.

Output

Выведите ответ на задачу.

Examples

standard input	standard output
1 0 0	2
4 0 0 1 0 0 1 -1 -1	15

Problem I. Декартово дерево

Input file: *standard input*
Output file: *standard output*
Time limit: 0.5 seconds
Memory limit: 256 mebibytes

По множеству точек на плоскости (с различными координатами x и различными координатами y) можно единственным образом построить декартово дерево (если рассматривать x как ключ, а y как приоритет). Мальчик по очереди ставит n точек равномерно в квадрат 1×1 и добавляет их в декартово дерево (изначально пустое). Будем говорить, что произошла перестройка, если добавленная вершина не является листом в новом дереве. Требуется посчитать мат. ожидание числа перестроек.

Поскольку вероятность совпадения каких-либо двух x или y равна нулю, можно считать, что они все различны.

Input

В единственной строке записано число n ($1 \leq n \leq 10^9$).

Output

Выведите ответ на задачу с абсолютной или относительной погрешностью не более 10^{-9}

Examples

standard input	standard output
2	0.5000000000
5	2.2388888888

Note

Декартово дерево — это структура данных, объединяющая в себе бинарное дерево поиска и бинарную кучу.

Более строго, это бинарное дерево, в узлах которого хранятся пары (x, y) , где x — это ключ, а y — это приоритет. Оно является двоичным деревом поиска по x и пирамидой по y . Предполагая, что все x и все y являются различными, получаем, что если некоторый элемент дерева содержит (x_0, y_0) , то у всех элементов в левом поддереве $x < x_0$, у всех элементов в правом поддереве $x > x_0$, а также и в левом, и в правом поддереве имеем: $y < y_0$. (Определение из Викиконспектов ИТМО).

Problem K. Двудольный граф

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Дан двудольный граф. В первой доле содержится n вершин с индексами от 0 до $n - 1$, а во второй — m вершин с индексами от 0 до $m - 1$. Изначально проведено k рёбер. На i -ом (начиная с нулевого) шаге проводится ребро между вершиной из первой доли с индексом $(i \bmod n)$ и вершиной из второй доли с индексом $(i \bmod m)$.

После сколько шагов граф станет связным?

Input

В первой строке записаны три числа: n , m и k ($1 \leq n, m \leq 10^9$, $0 \leq k \leq 10^5$).

В следующих k строках записаны по два числа u_i и v_i — индексы вершин из первой и второй доли, соединённых ребром ($0 \leq u_i < n$, $0 \leq v_i < m$).

Возможны кратные рёбра.

Output

Выведите ответ на задачу. В случае, если граф никогда не станет связным, выведите -1 .

Examples

standard input	standard output
3 5 2 1 3 2 1	5
3 3 1 0 2	-1

Problem L. Count it!

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Даны два целых числа a и b , выпишем все числа от a до b в десятичном представлении на листок бумаги. Ваша задача — найти количество вхождений каждой цифры. Например, если $a = 104$ и $b = 112$, на бумаге будут выписаны числа 104 105 106 107 108 109 110 111 112, итого в ответе 7 нулей, 13 единиц, одна двойка, 0 троек, а все остальные цифры встречаются только один раз.

Input

Входной файл состоит из не более, чем 500 строк. Каждая строка содержит два целых числа a и b , где $0 < a, b < 10^8$.

Входной файл завершается строкой с $a = b = 0$, обрабатывать которую не следует.

Output

Для каждого тестового примера выведите 10 чисел, разделённых пробелом; i -е число в строке должно обозначать количество цифр $i-1$ (то есть первое — количество нулей, второе — количество единиц и так далее, последнее — количество девяток).

Example

standard input									
1	10								
104	112								
346	542								
0	0								
standard output									
1	2	1	1	1	1	1	1	1	1
7	13	1	0	1	1	1	1	1	1
40	40	40	93	136	82	40	40	40	40

Problem M. Game

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Алиса и Боб играют в следующую игру. Игра проходит на доске $1 \times N$. Первоначально каждое поле доски содержит фишку с написанным на ней целым числом, всего использованы N фишек, на которых написаны попарно различные целые числа от 1 до N включительно.

Игроки делают ходы по очереди, снимая по одной фишке с доски. Фишка может быть снята только в случае, если соседние клетки пусты или содержат фишки, на которых записаны числа, меньшие, чем на ней.

Выигрывает тот, кто снимает фишку с записанной на ней единицей. По стартовой позиции определите, кто из игроков выиграет при оптимальной игре, если Алиса ходит первой.

Input

Входной файл начинается строкой, содержащей одно целое число T , $1 \leq T \leq 100$ — количество тестовых примеров.

Каждый тестовый пример начинается строкой, содержащей одно целое число N , $1 \leq N \leq 100$ — длину доски. В следующей строке заданы N целых чисел от одного до N — числа, написанные на фишках и перечисленные слева направо (от первой клетки к N -й).

Output

Для каждого тестового примера выведите имя выигравшего игрока в отдельной строке.

Example

standard input	standard output
2	Bob
3	Alice
1 3 2	
4	
1 4 2 3	

Note

В первом тестовом примере Алиса может снять только тройку, после чего Боб снимает 1 и выигрывает (так как единственная соседняя с 1 клетка стала пустой).

Во втором тестовом примере Алиса снимает тройку, оставляя Боба в ситуации, подобной только что разобранный, но с его первым ходом.

Problem N. Notebooks

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Ценность ноутбука для потребителя, согласно исследованию маркетинговой компании «BadBook», определяется двумя параметрами — продуктивностью и компактностью.

Некоторые предпочитают большой и тяжёлый ноутбук с высокой производительностью, некоторые — лёгкий и тонкий ноутбук, даже если его производительность невысока.

Компания «Badbook» составила список ноутбуков, сопроводив каждую модель двумя числами от 0.0 до 10.0.

Первое число обозначает оценку производительности (чем выше, тем лучше для потребителя). Второе — оценку размера (чем ниже, тем лучше для потребителя).

По заданному списку ноутбуков найдите длину наибольшей подпоследовательности ноутбуков, каждый из которых строго лучше для потребителя по каждому из параметров, чем предыдущий.

Input

Первая строка входа содержит одно целое число T ($1 \leq T \leq 20$) — количество тестовых примеров.

Каждый тестовый пример начинается строкой, содержащей одно целое число N ($1 \leq N \leq 200$) — количество ноутбуков в списке. Далее следуют N строк, каждая из которых содержит два вещественных числа p_i и w_i ($0 \leq p_i, w_i \leq 10.0$) — оценку производительности и размера очередного ноутбука.

Output

Для каждого тестового примера выведите строку, содержащую длину наибольшей подпоследовательности ноутбуков, каждый из которых строго превосходит предыдущий по обоим параметрам.

Example

standard input	standard output
2	2
2	1
1.0 2.0	
2.4 0.0	
3	
3.1 3.1	
3.2 3.1	
3.3 3.1	

Problem O. Painter Robot

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Робот-художник устроен следующим образом. Робот представляет собой карандаш, касающийся бумаги, и процессор, выдающий последовательность команд. Бумага может быть представлена как бесконечная двумерная плоскость.

В начале карандаш позиционируется в исходное положение (X_0, Y_0) командой, состоящей из двух целых чисел X_0 и Y_0 . Каждая следующая команда состоит из двух целых чисел X_i и Y_i , и предписывает устройству с карандашом переместиться в точку (X_i, Y_i) , оставляя за собой след на бумаге.

Гарантируется, что никакие две соседние команды не совпадают и что последняя команда содержит числа X_0 и Y_0 .

Дополнительно оказалось, что никакие две линии, нарисованные роботом, не перекрываются (но могут пересекаться).

Найдите количество частей, на которые нарисованная роботом замкнутая линия разбивает плоскость.

Input

На вход подаётся не более 25 тестовых примеров.

Каждый тестовый пример начинается строкой, содержащей целое число $4 \leq N \leq 300$ — количество команд. Вторая строка содержит N пар целых чисел задают команды процессора. Гарантируется, что используемые в командах координаты не превосходят 300 по абсолютной величине.

Ввод завершается примером с $N = 0$, обрабатывать который не требуется.

Output

Для каждого тестового примера выведите одно целое число — требуемое количество частей.

Examples

standard input	
5	
0 0 0 2 2 2 2 0 0 0	
7	
0 0 0 4 1 0 1 4 4 0 2 4 0 0	
0	
standard output	
2	
5	

Problem P. Convex Polyhedron

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Для вписанного в сферу многогранника соблюдается следующее соотношение: значение $x = V - E + F$, где V задаёт количество вершин, E — количество рёбер и F — количество граней, постоянно.

По заданным V и E требуется найти F .

Input

Первая строка входа содержит целое число T ($1 \leq T \leq 100$) — количество тестовых примеров.

Каждый тестовый пример состоит из одной строки, содержащей два целых числа V и E ($4 \leq V, E \leq 100$) — количества вершин и количества рёбер, соответственно.

Output

Для каждого тестового примера в отдельной строке выведите ответ — количество граней.

Example

standard input	standard output
1	4
4 6	