

Problem A. ASCII Addition

Input file: *standard input*
 Output file: *standard output*
 Time limit: 1 second
 Memory limit: 512 mebibytes

Требуется сложить два целых числа, представленных в виде *ASCII art*.

ASCII art представляет собой матрицу высоты 7 из символов, при этом в матрице могут встречаться только точка и строчная латинская буква 'x'.

Дано выражение в форме $a + b$, где a и b — целые положительные числа в десятичной записи. Выражение записано в *ASCII art* следующим образом: все символы (цифры чисел a и b и знак '+') представляют собой матрицы 7×5 , после чего они объединяются в общую картинку так, что между любыми двумя соседними матрицами-символами вставляется колонка, состоящая из точек.

Вот матрицы для всех цифр и знака '+': are as follows:

```

xxxxx  . . . . x  xxxxxx  xxxxxx  x . . . x  xxxxxx  xxxxxx  xxxxxx  xxxxxx  . . . . .
x . . . x  . . . . x  . . . . x  . . . . x  x . . . x  x . . . .  x . . . .  . . . . x  x . . . x  x . . . x  . . x . .
x . . . x  . . . . x  . . . . x  . . . . x  x . . . x  x . . . .  x . . . .  . . . . x  x . . . x  x . . . x  . . x . .
x . . . x  . . . . x  xxxxxx  xxxxxx  xxxxxx  xxxxxx  xxxxxx  . . . . x  xxxxxx  xxxxxx  xxxxxx  xxxxxx
x . . . x  . . . . x  x . . . .  . . . . x  . . . . x  . . . . x  x . . . x  . . . . x  x . . . x  . . . . x  . . x . .
x . . . x  . . . . x  x . . . .  . . . . x  . . . . x  . . . . x  x . . . x  . . . . x  x . . . x  . . . . x  . . x . .
xxxxxx  . . . . x  xxxxxx  xxxxxx  . . . . x  xxxxxx  xxxxxx  . . . . x  xxxxxx  xxxxxx  . . . . .
    
```

По заданному выражению в *ASCII art* виде найдите сумму и также запишите её в *ASCII art* в соответствии с вышеуказанными правилами.

Input

Вход состоит ровно из 7 строк и содержит *ASCII art* запись выражения в форме $a + b$, где a и b — целые положительные числа, записываемые не более, чем 9 десятичными цифрами без ведущих нулей.

Output

Выведите 7 строк, содержащие *ASCII art* запись суммы (тоже без ведущих нулей).

Example

standard input
<pre> x . xxxxxx . xxxxxx . x . . . x . xxxxxx . xxxxxx . xxxxxx xxxxxx . xxxxxx . xxxxxx x x x . x . . . x . x x x x x . x . . . x . x . . . x x x x . x . . . x . x x x x . x . . . x . x . . . x x . xxxxxx . xxxxxx . xxxxxx . xxxxxx . xxxxxx x . xxxxxx . xxxxxx . xxxxxx . x . . . x x . x x x x x x x x . x . . . x x . x x x x x x x x . x . . . x x . xxxxxx . xxxxxx x . xxxxxx . xxxxxx x xxxxxx . xxxxxx . xxxxxx </pre>
standard output
<pre> x . xxxxxx . xxxxxx . xxxxxx . x . . . x . xxxxxx . xxxxxx x x x . x x x x x x . x x x x . xxxxxx . xxxxxx . xxxxxx . xxxxxx . xxxxxx x x . x x x x x x . x x x x x x . xxxxxx . xxxxxx . xxxxxx x . xxxxxx x </pre>

Problem B. Book Borders

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

Некоторая книга печатается с использованием равношириного шрифта и простого жадного алгоритма для заполнения каждой строчки. Содержание книги представляет собой последовательность слов, каждое из слов содержит не менее одного символа.

Перед печатью мы выбираем *максимальную длину строки* и обозначаем это значение за m . Каждая строка может быть не более m символов в длину (включая и пробелы между словами). Алгоритм обрабатывает слова одно за другим и печатает каждое слово, вставляя между соседними словами в строке ровно один пробел. Если слово «вылезает» за m символов хотя бы одной буквой, то оно печатается с начала следующей строки.

```
|its.a.long...|           |its.a.long.way|  
|way.to.the...|         |to.the.top.if.|  
|top.if.you...|         |you.wanna.rock|  
|wanna.rock.n.|         |n.roll.....|  
|roll.....|
```

Text from the example input with maximum line lengths 13 and 14

Вам задан текст, который надо напечатать. Вы экспериментируете с различными значениями максимальной длины строки m . Для фиксированного m *последовательность первых слов* — это последовательность, которая состоит из первых слов каждой строки, перечисленных в порядке от верхней строки к нижней. В случае, указанном в иллюстрации, при $m = 14$ соответствующая последовательность — “its to you n”.

По заданному тексту и двум целым числам a и b найдите длину последовательности первых слов для каждого m между a и b включительно. Длина последовательности — сумма длин всех составляющих её символов (в том числе и пробелов).

Input

Первая строка входа содержит текст, который надо напечатать — последовательность слов, разделённых ровно одним пробелом. Каждое слово представляет собой строку, состоящую из одной или более строчной латинской буквы.

Вторая строка содержит два целых числа a и b — границы интервала, в котором выбирается m .

Гарантируется, что $1 \leq w \leq a \leq b \leq z \leq 500\,000$, где w — длина самого длинного слова в тексте и z — общее количество символов (включая пробелы) в тексте.

Output

Выведите $b - a + 1$ строку; k -я из этих строк должна содержать одно целое число — длину последовательности первых слов при $m = a - 1 + k$.

Example

standard input	standard output
its a long way to the top if you	22
wanna rock n roll	12
13 16	12
	15

Problem C. Counting Cities

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

Карл часто ездит в командировки. Каждая командировка представляет собой посещение ровно одного города.

Сейчас новый начальник Карла хочет узнать, во сколько различных городов Карл съездил в командировки. Босс попросил по списку командировок Карла посчитать, в каком количестве городов Карл был хотя бы однажды.

Input

Первая строка входа содержит целое положительное число $T \leq 50$ — количество тестовых примеров.

Первая строка каждого тестового примера содержит одно целое положительное число n ($1 \leq n \leq 100$), обозначающее количество поездок Карла в командировки. Каждая из последующих n строк содержит одно непустое слово, состоящее из строчных латинских букв и имеющее длину не более 20 — имя города, в который ездил Карл во время очередной командировки.

Output

Для каждого тестового примера выведите одну строку, содержащую количество различных городов, которые Карл посетил за время командировок.

Examples

standard input	standard output
2	4
7	1
zagreb	
krakow	
warsaw	
krakow	
prague	
zagreb	
krakow	
3	
barnaul	
barnaul	
barnaul	

Problem D. Digit Division

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

Дана последовательность из n десятичных цифр. Требуется разделить эту последовательность на одну или несколько непрерывных подпоследовательностей так, чтобы каждая такая последовательность, интерпретированная как целое число, делилась бы на заданное целое число m .

Найдите остаток от количества различных разбиений от деления на $10^9 + 7$. При определении того, что разбиения раздельны, мы считаем два разбиения, отличающиеся только границами между подпоследовательностями (например, $2|22$ и $22|2$) различными.

Input

Первая строка входа содержит два целых числа n и m ($1 \leq n \leq 300\,000$, $1 \leq m \leq 1\,000\,000$) — длину последовательности и значение делителя. Во второй строке задана сама последовательность из n подряд идущих цифр.

Output

Выведите одно целое число — количество различных разбиений по модулю $10^9 + 7$.

Example

standard input	standard output
4 2 1246	4
4 7 2015	0
4 3 3003	8

Problem E. Electoral Estimato~~ns~~

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

В Байтландии стартовал первый тур президентских выборов. Так как в выборах участвует более двух кандидатов, то возможно, что победитель (кандидат, который получит наибольшее количество голосов) не наберёт большинства всех голосов. В этом случае будет проведён второй тур.

Для того, чтобы оценить предвыборные ожидания, байтландские учёные промоделировали голосование. Они попросили Вас написать программу, которая по количеству голосов, набранных кандидатами, определяет лидера голосования (в случае, если наибольшее количество голосов собрал один кандидат) и определяет, является ли его победа финальной или же потребуется второй тур.

Input

Первая строка входа содержит целое положительное число $T \leq 500$ — количество тестовых примеров.

Первая строка каждого тестового примера содержит целое положительное число n ($2 \leq n \leq 10$) — количество кандидатов. i -я из последующих n строк содержит целое неотрицательное число v_i ($0 \leq v_i \leq 50\,000$) — количество голосов, поданных за кандидата с номером i . Гарантируется, что в голосовании принимал участие хотя бы один избиратель.

Output

Для каждого тестового примера выведите “**Victory**” и через пробел — номер выигравшего кандидата, если победитель набрал более половины количества голосов и второй тур не потребуется. В противном случае выведите “**Leader**” и номер лидирующего кандидата, если ровно один кандидат набрал наибольшее количество голосов; если же таких кандидатов как минимум два, выведите “**Tie**”. Кандидаты пронумерованы последовательными целыми числами от 1 до n .

Example

standard input	standard output
4	Victory 2
3	Leader 1
10	Tie
21	Leader 4
10	
3	
20	
10	
10	
3	
10	
10	
10	
4	
15	
15	
15	
45	

Problem F. Funny Function

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

Фрэнк называет число *забавным*, если его запись в K -ичной системе счисления не содержит двух последовательных нулей.

Сейчас Фрэнк придумал свою собственную функцию F и назвал её *забавной функцией*. $F(N, K)$ равна количеству забавных N -значных чисел без ведущих нулей в K -ичной системе счисления.

По заданным N и K вычислите $F(N, K)$.

Input

Входная строка содержит два целых числа N и K — количество знаков и основание системы ($2 \leq K \leq 10$, $2 \leq N$, $N + K \leq 18$)

Output

Выведите $F(N, K)$.

Examples

standard input	standard output
3 6	175

Problem G. Gregor's Game

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

Грегор придумал новую карточную игру для детей. Правила этой игры очень просты: каждому игроку выдано некоторое количество карт. Каждая карта имеет картинку на каждой стороне. Игроки делают ходы по очереди и первый, у кого закончатся карты, является победителем.

Ход игрока состоит в выборе подмножества карт из его руки и выкладывании их на столе. Разрешено выкладывать любое количество карт с тем ограничением, что никакие две из выкладываемых игроком карт не лежат одинаковыми картинками вверх.

Помогите Грегору определить, может ли он выложить все свои карты.

Input

Первая строка входа содержит одно целое положительное число T ($T \leq 10$) — количество тестовых примеров.

Первая строка каждого тестового примера содержит одно целое число n , обозначающее количество карт у Грегора ($1 \leq n \leq 50\,000$). Далее следуют n строк, каждая из которых задаёт одну карту.

Картинки на картах заданы целыми числами. i -я карта задаётся двумя целыми числами p_i и q_i ($1 \leq p_i, q_i \leq 2n$).

Output

Для каждого тестового примера выведите в отдельной строке слово “Yes”, если Грегор сможет выложить на стол все свои карты, и “No” в противном случае.

Example

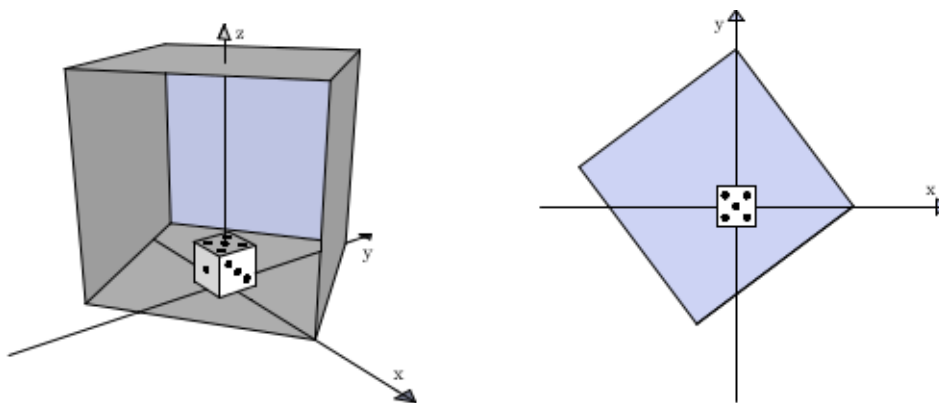
standard input	standard output
3	Yes
3	No
1 2	Yes
1 3	
2 3	
3	
1 2	
1 2	
1 2	
1	
1 1	

Problem H. Hovering Hornet

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

Вы поймали шмеля в коробку, которая лежит на Вашем обеденном столе. К сожалению, в этой же коробке оказалась и Ваша игральная кость, так что продолжить партию в настольную игру Вы не сможете, опасаясь быть ужаленными. Вместо игры Вы решили развлечься, вычислив ожидаемое количество кружков на кости, видимых шмелю.

Шмель, кость и коробка находятся в стандартном трёхмерном декартовом координатном пространстве, в котором координата x растёт на восток, координата y — на север и координата z — вверх. Поверхность стола соответствует плоскости $x-y$.



Perspective and the birds-eye view of the second example input

Игральная кость представляет собой куб $1 \times 1 \times 1$, расположенный на столе так, чтобы центр нижней стороны попал в начало координат. При этом координаты двух противоположных вершин равны $(-0.5, -0.5, 0)$ и $(0.5, 0.5, 1)$. На верхней стороне кости 5 очков, на южной — одно очко, на восточной — 3 очка, на северной — 6 очков, на западной — 4 очка и на нижней, невидимой — 2 очка.

Коробка представляет собой куб $5 \times 5 \times 5$, также стоящий на поверхности стола. Коробка задаётся координатами нижней грани — квадрата 5×5 .

Считая, что шмель летает по поверхности куба, двигаясь как равномерно распределённая случайная точка в непрерывном пространстве внутри куба, не занятом костью. Кость непрозрачна, таким образом, шмель видит кружок, только если отрезок, соединяющий центр кружка и положение шмеля, не пересекает внутренность кости.

Input

Вход состоит из четырёх строк. k -я строка содержит два вещественных числа x_k и y_k ($-5 \leq x_k, y_k \leq 5$) — координаты k -го угла коробки на плоскости $x-y$. Координаты перечислены в порядке обхода против часовой стрелки и они задают квадрат с длиной стороны, равной ровно 5.

Коробка полностью содержит кость; поверхность кости имеет общие точки только с нижней гранью коробки.

Output

Выведите одно вещественное число — ожидаемое количество видимых точек. Решение будет считаться верным, если абсолютная или относительная погрешность будет не хуже 10^{-6} .

Example

standard input	standard output
-2.5 -1.5 2.5 -1.5 2.5 3.5 -2.5 3.5	10.6854838710
3 0 0 4 -4 1 -1 -3	10.1226478495

Problem I. Ice Igloos

Input file: *standard input*
Output file: *standard output*
Time limit: 10 seconds
Memory limit: 512 mebibytes

Посёлок рыбаков на льду замёрзшего озера на крайнем Севере оказался под угрозой из-за необычно тёплой зимы — на поверхности озера стали образовываться трещины.

Посёлок состоит из n иглу сферической формы, каждое иглу занимает на поверхности озера круглый участок.

Иглу можно представить как круг на координатной плоскости: центром круга является точка с целыми координатами, а радиусом — положительное вещественное число r , меньшее 1, такое, что $10r$ является целым числом.

По заданным местоположениям трещин рыбаки хотят знать, сколько иглу находятся под угрозой от каждой трещины. Более формально, задано q запросов, представляющих собой отрезок прямой, заданный двумя концами; требуется найти количество кругов, пересекаемых каждой прямой. Круг и прямая пересекаются, если прямая имеет хотя бы одну общую точку со внутренностью круга.

Input

Первая строка входа содержит целое число n ($1 \leq n \leq 100\,000$) — количество иглу. Каждая из последующих n строк содержит три числа x , y и r — координаты центра и радиус одного иглу. Координаты x и y — целые числа, такие, что $1 \leq x, y \leq 500$, а r — вещественное число, такое, что $10 \cdot r$ является целым и $0 < r < 1$. Никакие два иглу не пересекаются и не касаются.

Следующая строка содержит целое число q ($1 \leq q \leq 100\,000$) — количество запросов. Каждая из последующих q строк содержит четыре целых числа x_1, y_1, x_2, y_2 ($1 \leq x_1, y_1, x_2, y_2 \leq 500$) — координаты концов отрезка, задающего трещину. Отрезок имеет ненулевую длину. Концы отрезков могут быть внутри иглу.

Гарантируется, что для каждого иглу i и отрезка s квадрат расстояния между s и серединой i или меньше, чем $r^2 - 10^{-5}$, или больше, чем $r^2 + 10^{-5}$, где r — радиус соответствующего иглу.

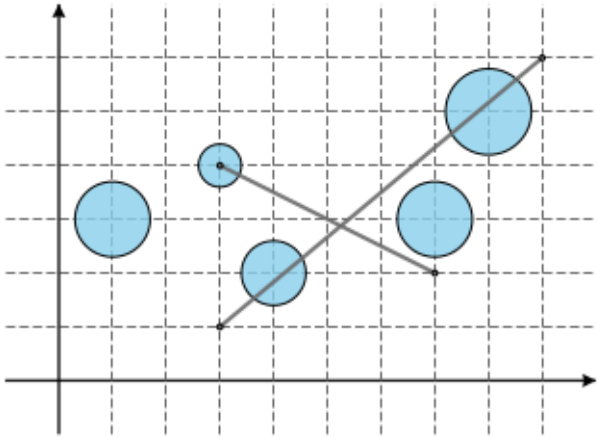
Output

Выведите q строк; k -я из них должна содержать одно целое число — количество иглу, которые пересекаются с k -м отрезком.

Example

standard input	standard output
5	2
4 2 0.6	1
7 3 0.7	
8 5 0.8	
1 3 0.7	
3 4 0.4	
2	
3 1 9 6	
3 4 7 2	

Note



Problem J. Juice Junctions

Input file: *standard input*
Output file: *standard output*
Time limit: 4 seconds
Memory limit: 512 mebibytes

Вам поручено усовершенствовать систему транспортировки апельсинового сока на одном фруктоперерабатывающем заводе. Система состоит из труб и коллекторов. Все трубы являются двунаправленными и имеют одинаковую пропускную способность, равную одному литру в секунду. Трубы могут соединяться только в коллекторах, причём каждый коллектор соединяет не более трёх труб. Пропускная способность коллектора нелимитирована. Коллекторы занумерованы целыми числами от 1 до n .

Перед тем, как предложить усовершенствования, Вам требуется проанализировать существующую систему. Для двух различных коллекторов s и t определим $s - t$ поток как наибольший объём сока (в литрах в секунду), который может идти через систему, если источник установлен в коллекторе s , а приёмник — в источнике t . Например, для системы из первого примера к задаче, поток 1-6 равен 3, а поток 1-2 равен 2.

Найдите сумму $a - b$ потоков по всем парам коллекторов a и b , таких, что $a < b$.

Input

Первая строка входа содержит два целых числа n и m ($2 \leq n \leq 3000$, $0 \leq m \leq 4500$) — количество коллекторов и количество труб. Каждая из последующих m строк содержит два различных целых числа a и b ($1 \leq a, b \leq n$), задающих трубу, соединяющую коллекторы a и b .

Каждый коллектор не может быть соединён более, чем с тремя коллекторами. Каждая пара коллекторов соединена не более, чем одной трубой.

Output

Выведите одно целое число — сумму всех $a-b$ потоков для всех пар коллекторов a и b , таких, что $a < b$.

Examples

standard input	standard output
6 8 1 3 2 3 4 1 5 6 2 6 5 1 6 4 5 3	36
4 2 1 2 3 4	2

Note

Иллюстрация к первому примеру:

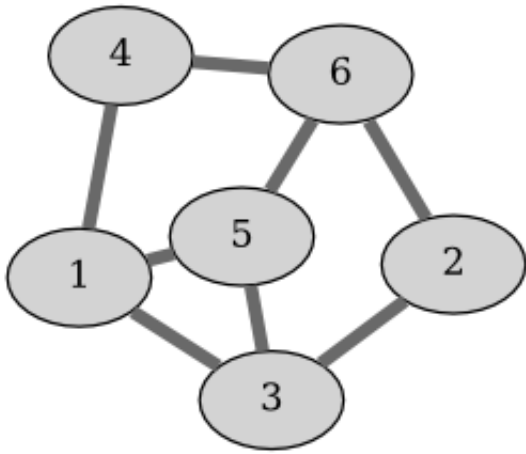
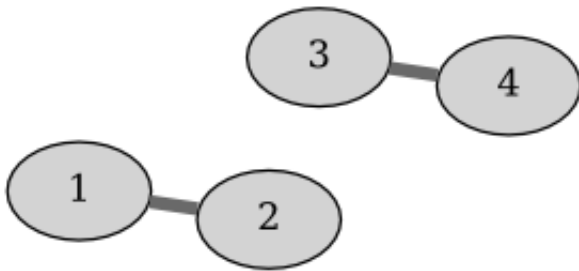


Иллюстрация ко второму примеру:



Problem K. Kernel Knights

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

$2n$ рыцарей из двух враждующих кланов встретились на крупном рыцарском турнире. Сразу по прибытии каждый рыцарь вызвал на поединок какого-то рыцаря из другого клана.

Назовём *ядром* турнира такое подмножество рыцарей S , в котором

- Ни один рыцарь из S не был вызван на поединок другим рыцарем из S .
- Каждый рыцарь извне S был вызван на поединок каким-то рыцарем из S .

Вам задан список вызовов на поединок, найдите хотя бы одно ядро. Гарантируется, что оно всегда существует.

Input

Первая строка теста содержит одно целое число n ($1 \leq n \leq 100\,000$) — количество рыцарей в каждом клане. Рыцари первого клана пронумерованы последовательными целыми числами от 1 до n , рыцари из второго клана — последовательными целыми числами от $n + 1$ до $2n$.

Следующая строка содержит целые числа f_1, f_2, \dots, f_n ; k -е из этих чисел, f_k , является номером рыцаря, вызванного на поединок рыцарем k ($n + 1 \leq f_k \leq 2n$).

Следующая строка содержит целые числа s_1, s_2, \dots, s_n ; k -е из этих чисел, s_k , является номером рыцаря, вызванного на поединок рыцарем $n + k$ ($1 \leq s_k \leq n$).

Output

Выведите номера рыцарей, составляющих ядро, в одной строке. Если решений более одного, выведите любое.

Example

standard input	standard output
4 5 6 7 7 1 3 2 3	1 2 4 8

Problem L. Larry's Lemma

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

В своём сочинении по нумерологии о свойствах магического числа 3 юный маг Ларри Лоттер упомянул следующую лемму: почти все целые числа могут быть представлены как число, заканчивающееся на 3 при выборе соответствующей системы счисления, для некоторых чисел такую систему можно выбрать более, чем одним способом — например, 11 представляется как 13_8 или 23_4 .

Преподаватель, чтобы проверить, что лемма имеет место быть, попросил Вас написать программу, которая найдёт наименьшее основание системы счисления для заданного числа такое, что его запись в системе с этим основанием будет заканчиваться на 3.

Input

Каждая строка входа содержит одно целое неотрицательное число n . Значение $n = 0$ обозначает конец ввода и не должно обрабатываться. Все числа на входе не превосходят 2^{31} , всего во входе не более 1000 ненулевых чисел.

Output

Для каждого ненулевого значения n во входе выведите в отдельной строке наименьшее основание системы счисления, в которой запись данного числа заканчивается на 3. Если такого основания не существует, выведите "Fail".

Example

standard input	standard output
11	4
123	4
104	101
2	Fail
3	4
0	