Problem A. Lucky Controller

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

Джон работает чиновником в визовом отделе посольства США по работе с участниками соревнований по программированию. Каждый день он получает бланки заявлений, которые впоследствии раздаёт участникам. Однако, согласно секретной инструкции Госдепартамента США, в первую очередь должны быть рассмотрены заявления, написанные на бланке со «счастливыми» номерами. Каждый номер бланка состоит из $2n$ цифр. Номер бланка называется «счастливым», если сумма первых n цифр равна сумме последних n цифр.

Джон знает, что ему выдадут на день k бланков. Также у него есть информация, что номера бланков идут подряд, а первый номер в последовательности бланков с равной вероятностью может быть любым числом из интервала от a до b включительно.

Джона интересует, каково матожидание количества заявлений со «счастливыми» номерами, которое ему будет выдано на следующий день.

Input

В единственной строке входного файла содержатся три целых числа a, b и k ($0 \le a \le b < 10^{12}, 1 \le k \le 10^5$). Числа a, b и $b + k$ состоят из одного и того же чётного количества цифр, не превосходящего 12 — количества цифр в номере каждого заявления. a и b могут начинаться с нулей.

Output

Выведите матожидание количества «счастливых» номеров заявлений в выданных на день бланках в виде несократимой дроби. Если ответ целый, то дробную черту выводить не нужно.

Example

standard input	standard output
0123 4567 150	6519/635
10 10 20	2
4000 4999 11	103/125

Problem B. Feed the Robot

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

Робот двигается по ленте, состоящей из $n + 1$ ячеек. Ячейки пронумерованы от 0 до n. Изначально робот находится в ячейке с номером 0. В каждой из остальных ячеек расположено некоторое количество кристаллов. Оказавшись в ячейке, робот забирает все находившиеся в ней кристаллы. Робот может сделать m прыжков в соседнюю ячейку и k прыжков через ячейку, при этом $m + 2k = n$. Робот может прыгать только вперёд.

По заданному расположению кристаллов на ленте вычислите, какое максимальное количество кристаллов сможет собрать робот.

Input

Первая строка входного файла содержит 3 целых числа n ($1 \le n \le 100$), m ($0 \le m \le 100$), k ($0 \le k \le 100$). Во второй строке заданы n целых чисел — количество кристаллов (не боле 100) в соответствующей ячейке ленты.

Output

В первой строке выходного файла должно быть выведено одно число — максимальное количество кристаллов. Вторая должна содержать $m + k + 1$ целых чисел — номера ячеек, которые посетил робот, начиная с ячейки с номером 0.

Example

standard input	standard output
5 1 2 5 2 7 3 1	13 0 1 3 5

Problem C. Collisions

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

На прямой расположены одинаковые шарики, которые могут двигаться только вдоль этой прямой. Изначально каждый шарик двигается с постоянной скоростью (знак скорости задаёт направление). После столкновения шарика A, двигающегося со скоростью V_A и шарика B, двигающегося со скоростью V_B шарик B начинает двигаться со скоростью V_A, а шарик A — со скоростью V_B. Вычислите общее количество произошедших при этом столкновений.

Input

В первой строке входного файла задано количество шариков N ($3 \le N \le 2 \cdot 10^5$). Каждая из последующих N строк содержит 2 целых числа — начальную координату и скорость соответствующего шарика. Все стартовые координаты находятся в диапазоне $-10^{11} < x_i < 10^{11}$, все скорости в диапазоне $-10^8 < v_i < 10^8$. Для любых двух шариков стартовые координаты различны; также гарантируется отсутствие «тройных» и более столкновений.

Output

Выведите одно целое число — общее количество столкновений или 987654321987654321, если количество столкновений бесконечно.

Example

standard input	standard output
3 -5 3 0 -1 7 -2	3

Problem D. Pairs

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	64 Mebibytes

На дипломатическом приёме находятся N дипломатов, причём число N чётно. Каждый дипломат ведёт беседу ровно с одним другим дипломатом. Разведке одной из стран удалось получить фотографии с приёма. Изучив эти фотографии, аналитики установили координаты каждого из дипломатов и попытались восстановить, кто из них с кем беседует. Алгоритм восстановления следующий: из дипломатов, ещё не распределённых по парам, выбираются двое, расстояние между которыми минимально, и принимается, что они беседуют между собой. В случае, если существует несколько пар дипломатов, в которых расстояния минимальны, выбирается пара с лексикографически наименьшим номером (все дипломаты пронумерованы по номерам их досье в разведке от 1 до N, внутри пары собеседников дипломат с наименьшим номером указывается первым).

Вам поручено реализовать этот алгоритм.

Input

Первая строка входного файла содержит чётное число N ($2 \le N \le 300$). В последующих N строках заданы координаты x_i и y_i каждого дипломата, причём в i-й из этих строк заданы координаты дипломата с номером досье i. Для двух различных дипломатов как минимум одна из соответствующих координат различна. Координаты не превосходят по абсолютной величине 10^8.

Output

Выведите $N/2$ строк — номера пар дипломатов, беседующих между собой. Пары выводятся в лексикографическом порядке, внутри каждой пары первое число должно быть меньше второго.

Examples

standard input	standard output
6 0 2 3 2 0 0 1 0 0 -2 2 -2	1 2 3 4 5 6
4 0 0 1 1 0 1 1 0	1 3 2 4

Problem E. Embassy

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	64 Mebibytes

Перед финалом крупного международного турнира по программированию участники со всех регионов Украины съехались в Киев, чтобы в американском посольстве получить визу. Как мы знаем из первой задачи, в американских посольствах с участниками соревнований по программированию работает ровно один чиновник, так что ничего удивительного, что образовалась огромная очередь из участников. Собеседование чиновника с каждым участником длится ровно час. Участники взяли билеты из Киева заранее и сейчас некоторые из них могут не успеть на поезд из-за того, что им придётся стоять в очереди. Спонсоры турнира готовы компенсировать участникам, не успевшим на поезд, затраты на обмен билета.

Ваша задача — расставить финалистов в очередь так, чтобы затраты спонсора были минимальны.

Input

Пусть N — количество финалистов, i — номер, под которым финалист зарегистрирован в системе, d_i — наиболее позднее время начала собеседования, после которого i-й финалист ещё может успеть на поезд, w_i — стоимость обмена билета для i-го финалиста.

В первой строке входного файла задано N, в N следующих строках заданы соответственно d_i и w_i (в i-й из этих строк).

Все числа во входном файле целые, положительные и не превышают 30000.

Output

Выведите оптимальную с точки зрения затрат спонсора очередь. В i-й строке вывода должен быть выведен номер участника, стоящего в очереди на i-м месте. Если таких очередей несколько, можно вывести любую из них.

Example

standard input	standard output
7 3 40 2 60 6 10 1 30 4 70 4 50 4 20	2 1 5 6 3 4 7

Problem F. Dictionary of Obscene Words

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

Дан текст T и список нецензурных слов S_1, S_2, ..., S_n. Требуется выяснить, содержит ли текст в качестве подпоследовательности одно из нецензурных слов Если содержит, найдите наименьший префикс текста T, содержащий эту подпоследовательность.

Input

Первая строка входного файла содержит одно целое число n — количество нецензурных слов в списке. Последующие n строк содержат слова из списка по одному в строке. Следующая строка содержит текст T. Суммарная длина слов в словаре не превышает 100 KiB (100×2^{10} *bytes*). Общий размер входного файла не превышает 1 MiB (2^{20} *bytes*). Список слов и текст состоят из символов с кодами от 32 до 127 включительно.

Output

Выведите `'NO'`, если в тексте не встречается нецензурных слов из списка. В другом случае выведите `'YES <X>'`, где X — длина наименьшего префикса текста T, содержащего какое-то нецензурное слово в качестве подпоследовательности.

Examples

standard input	standard output
2 hello world abracadabra	NO
2 hello world zzzheluuuulottt	YES 12

Problem G. Greatest Product

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	64 Mebibytes

Для каждого целого положительного числа x определим функцию $P(x)$, равную произведению цифр в десятичной записи числа x. По заданному N вычислить максимальное значение $P(x)$ при $x \le N$.

Input

Во входном файле содержится одно целое число N ($1 \le N \le 2 \cdot 10^9$).

Output

Выведите максимальное значение функции $P(x)$ на промежутке от 1 до N.

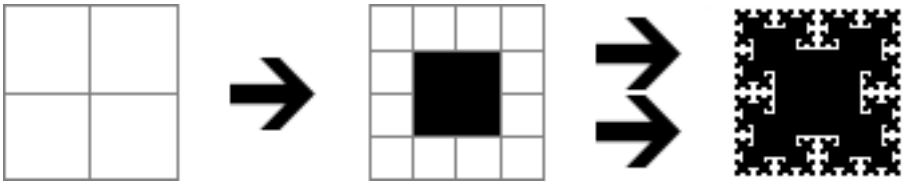
Examples

standard input	standard output
1	1
101090000	43046721
28994	10368
4876	2268
2789	1008

Problem H. Birthday Cake

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

Сегодня Федя празднует свой день рождения, и, пока гости не пришли, он украшает праздничный торт шоколадным кремом, но по-особенному. Вначале у Феди есть квадратный торт, разделенный на 4 равных квадрата из коржей белого цвета. Федя называет «фрактализацией» следующее действие — группируем все маленькие квадраты торта в группы 2×2 так, чтобы не осталось несгруппированных фрагментов, после чего каждый маленький квадрат делим на 4 равных квадрата и заполняем шоколадом 4 квадратика в середине каждой группы. Такое действие повторяется последовательно N раз. Иллюстрация снизу показывает первую «фрактализацию» и торт после пятой:



Теперь Федя хочет нарезать гостям куски торта с красивыми узорами, но ему трудно, смотря на торт в целом, оценить узор его части. Напишите программу, которая покажет узор заданной прямоугольной части торта.

Input

В единственной строке записаны пять целых неотрицательных чисел N, R_1, R_2, C_1, C_2. N — количество итераций вышеописанных изменений, проведённых над тортом ($N < 20$), R_1 и R_2 — это соответственно начальная и конечная строки, а C_1 и C_2 — начальный и конечный столбцы вырезанного куска. Действуют ограничения: $R_1 \le R_2$, $C_1 \le C_2$; $0 \le R_2 - R_1, C_2 - C_1 < 100$; $0 \le R_1, R_2, C_1, C_2 < 2N + 1$.

Output

Ожидается $R_2 - R_1 + 1$ строк по $C_2 - C_1 + 1$ символов в каждой. Если соответствующий квадратик залит шоколадом, следует вывести 1, в противном случае выведите 0.

Examples

standard input	standard output
1 0 3 0 3	0000 0110 0110 0000
2 0 3 0 3	0000 0110 0111 0011
13 50 55 95 100	101111 100111 100111 101111 101101 100001

Problem I. Hexaroman Numbers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

Рассмотрим традиционную «римскую» систему счисления и построим её обобщение.

В обычной («десятичной») римской записи используются 7 римских цифр: `I` = 1, `V` = 5, `X` = 10_{10}, `L` = 50_{10}, `C` = 100_{10}, `D` = 500_{10}, `M` = 1000_{10}.

Обычно римские цифры записываются в убывающем порядке и при этом значения цифр складываются (например, `MMX` (2010) интерпретируется как $1000 + 1000 + 10$). Если цифра с меньшим значением идёт перед цифрой с большим значением, меньшее значение вычитается из большего и прибавляется к общей сумме. Например, `MCMXLIV` равно 1944. На конструкции с вычитанием существует следующее ограничение: `I` может идти только перед `V` and `X`, `X` может идти только перед `L` or `C`. За цифрами `V`, `L`, and `D` могут идти только цифры с большим значением.

Введём шестнадцатеричную римскую запись по следующим правилам: Цифрам присваиваются значения `I` = 1, `V` = 8, `X` = 10_{16}, `L` = 80_{16}, `C` = 100_{16}, `D` = 800_{16}, `M` = 1000_{16}.

Ограничения на использование цифр такие же, как в «десятичной» римской системе, за исключением того, что запись типа `IIX` является корректной. Если какое-то число из-за этого может быть представлено разными способами, используется запись с меньшим количеством знаков, если количество знаков одинаково, то используется запись с меньшим количеством вычитаний (например, из записей `IIIIX` и `VIIII` для числа C_{16} выбирается вторая) Например, число F_{16} записывается как IX_{16}, а $5C8_{16}$ — как $CCCDLXXXXV_{16}$.

Напишите программу, которая производит операции над шестнадцатеричными римскими цифрами (сложение, вычитание, умножение). Гарантируется, что входные данные и результаты вычислений — целые положительные числа, не превышающие $4FFF_{16}$.

Input

В первой строке входного файла задано целое число N ($0 < N \leq 100$) — количество тестовых примеров. Каждый тестовый пример содержит задание в формате `<A><O><B>`. `<A>` и `<B>` — числа в шестнадцатеричной римской записи, `<O>` — одна из операций: `+`, `-`, `*`. Ни один тестовый пример не содержит пробелов.

Output

Для каждого примера выведите в отдельной строке результат вычислений, представленный в шестнадцатеричной римской записи.

Example

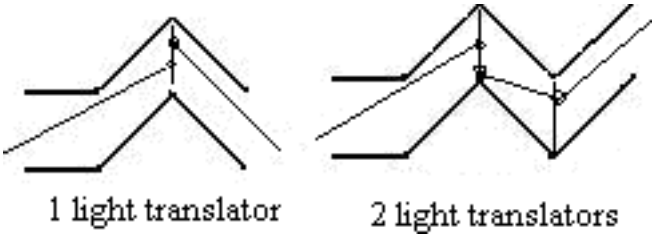
standard input	standard output
3 XIIV+XXXXII XXIIII*XXXIII XXIV*IV	XXXLV CDXXVIIII CXI

Problem J. Beam in Tunnel (Div 1 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

Туннель с квадратным сечением состоит из $(n-1)$ секций. Пол в каждой секции плоский и может быть наклонён вверх или вниз. Координаты (x_1, y_1), (x_2, y_2), ..., (x_n, y_n) ($x_1 < x_2 < \ldots < x_n$) описывают точки, где пол туннеля начинается, заканчивается (первая и последняя пара координат соответственно), или где секции соединяются. Высота туннеля равна 1, то есть потолок i-й «точки соединения» имеет координаты $(x_i, y_i + 1)$.

В туннель направляется луч лазера. Для того, чтобы обеспечить передачу сигнала, могут быть использованы преобразователи света, которые можно установить на границах секций. Эти преобразователи перенаправляют луч в нужном направлении, причём не обязательно из точки падения луча (см. иллюстрацию).



Так как преобразователи перекрывают туннель целиком и могут быть установлены только на границах секций, то каждый преобразователь однозначно задаётся своей координатой — одним из чисел x_1, x_2, ..., x_n.

Определите минимальное количество преобразователей, необходимых для того, чтобы свет дошёл до конца туннеля. Касание лучом стенок туннеля не допускается.

Input

Первая строка входного файла содержит целое число N ($2 \le N \le 1000$). Последующие N строк содерждат 2 вещественных числа, заданных в типе extended — координаты x_i и y_i ($-10000 \le x_i, y_i \le 10000$).

Output

Выведите одно число — наименьшее количество преобразователей, необходимое для того, чтобы луч дошёл до конца туннеля.

Examples

standard input	standard output
4 1 1 2 1 3 2 4 1	1
5 1 1 2 1 3 2 4 1 5 2	2

Problem K. Key

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

Как известно, хакерам часто приходится взламывать пароли для различных систем шифрования данных. Так и перед начинающим хакером Биллом однажды возникла такая задача. Проведя несколько экспериментов, он заметил определённые закономерности в формировании ключа. Ему известно, что ключ — это натуральное нечётное число K, такое, что $(K-1)!$ не делится на K^2 и принимает значения из диапазона между A и B ($A \le K \le B$). (Напомним, что $(K-1)! = (K-1) \cdot (K-2) \cdot \cdots \cdot 2 \cdot 1$). А дальше дело пошло хуже, так как Билл не силён в математике. Чтобы помочь юному хакеру, вы должны по заданным ограничениям вывести все возможные значения ключа.

Input

Вход состоит из двух целых чисел A и B ($3 \le A < B \le 10^{18}$, $(B-A) \le 100$).

Output

Ваша программа должна вывести на стандартный вывод через пробел все возможные ключи K, которые удовлетворяют вышеперечисленным условиям. Гарантируется, что хотя бы один такой ключ существует.

Examples

standard input	standard output
3 10	3 5 7 9
7 14	7 9 11 13

Problem L. Game on Chessboard (Div 1 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 Mebibytes

Есть шахматная доска размера $M \times N$. На ней в некоторых клетках расположены K вымышленных шахматных фигур, называемых (p, q)-скакунами $(p < q)$. Ход скакуна похож на ход обычного шахматного коня. При своем ходе скакун перемещается на p клеток по горизонтали и на q по вертикали или на q клеток по горизонтали и на p по вертикали. При этом перемещение на q должно выполняться обязательно либо вверх, либо влево (то есть в сторону уменьшения соответствующей координаты). Недопустим ход, который выводит фигуру за пределы доски, однако в одной клетке могут находиться несколько фигур.

Два игрока играют в игру. Они ходят по очереди. В свой ход игрок обязан выбрать одного из скакунов и выполнить им допустимый ход. Проигрывает тот, кто не может сделать ход.

Определите кто выиграет, предполагая что оба игрока играют оптимально,

Input

В первой строке входного файла задано 5 целых чисел M, N, K, p, q $(1leM, N \leq 10^9, 1 \leq K \leq 10^5, 1 \leq p < q \leq 20)$. В каждой из последующих K строк задаются координаты соответствующего скакуна r_i и c_i $(1 \leq r_i \leq M,\ 1 \leq ci \leq N)$.

Output

В выходной файл выведите `'First'`, если при оптимальной игре выигрывает первый игрок, и `'Second'` в противном случае.

Examples

standard input	standard output
10 10 2 1 2 3 7 7 3	Second
7 5 3 1 3 2 3 1 5 4 3	First