

Problem A. 2016

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

Число 2016 имеет очень много делителей.

Обозначим за $d(n)$ количество делителей числа n . Например, $d(12) = 6$, так как число 12 имеет 6 делителей: 1, 2, 3, 4, 6 и 12.

Целое положительное число x назовём *особенным*, если количество целых положительных чисел y , для которых одновременно $y < x$ и $d(y) > d(x)$ не больше одного. Например, 2016 является особенным, так как среди чисел, меньших 2016, только число 1680 имеет больше делителей, чем 2016.

Вам задано целое число K . Вычислите K -е наименьшее особенное число. Если это число превосходит 10^{18} , выведите -1 .

Input

Вход содержит одно целое число K ($1 \leq K \leq 10^9$).

Output

Выведите одно число — ответ к задаче.

Examples

standard input	standard output
10	14
1000000000	-1

Note

Наименьшими особенными числами являются 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, ...

Problem B. Airports

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

Snuke — владелец N аэропортов. Изначально между ними нет авиарейсов.

Координаты i -го аэропорта — (x_i, y_i) . Snuke выбирает константу D и для каждой пары аэропортов p и q добавляет авиарейс между ними, если манхэттенское расстояние между p и q не менее D . Найдите наибольшее D при котором из каждого аэропорта можно долететь в любой другой с пересадками или без них.

Напомним, что манхэттенское расстояние между двумя точками с координатами (x_1, y_1) и (x_2, y_2) определяется как $|x_1 - x_2| + |y_1 - y_2|$.

Input

Первая строка входа содержит одно целое число N ($2 \leq N \leq 10^5$). Далее следуют N строк, i -я из которых содержит по два целых числа x_i и y_i — координаты i -го аэропорта ($0 \leq x_i, y_i \leq 10^9$). Никакие два аэропорта не находятся в одной точке.

Output

Выведите ответ к задаче.

Example

standard input	standard output
6 1 7 8 5 6 3 10 3 5 2 6 10	9

Problem C. Jump

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

Snuke стоит на бесконечно длинной дороге.

Его положение на этой дороге определяется вещественным числом.

Snuke может делать N типов прыжков. Прыжок типа i симметричен относительно точки a_i . То есть если он прыгает из точки x , то он переместится в точку $2a_i - x$.

Вам задано Q запросов. В i -м запросе Вас просят вычислить наименьшее количество прыжков, за которое Snuke доберётся из точки s_i в точку t_i или вывести -1 , если добраться из s_i в t_i с помощью заданных прыжков невозможно.

Input

Первая строка входа содержит одно целое число N ($1 \leq N \leq 200$). Последующие N строк содержат целые числа a_i , по одному на строке ($0 \leq a_1 < \dots < a_N \leq 10^4$). Следующая строка содержит одно целое число Q — количество запросов ($0 \leq Q \leq 10^5$).

Каждая из последующих Q строк содержит один запрос и состоит из двух целых чисел s_i и t_i ($0 \leq s_i, t_i \leq 10^4$).

Output

Для каждого запроса выведите в отдельной строке ответ на него.

Example

standard input	standard output
4	-1
1	-1
2	2
4	2
7	-1
10	-1
2 3	0
5 6	3
6 0	1
3 7	0
10 3	
7 6	
5 5	
2 10	
4 10	
10 10	

Problem D. Merge

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

Snuke хочет создать массив R , объединив два массива P и Q . Формально массив R строится по следующим правилам:

- Первоначально массив R пуст.
- Пока как минимум один из массивов P и Q непуст, выбираем непустой массив (P или Q) и переносим из него самый левый элемент в правую часть массива R .

Вам заданы строки P и Q , которые являются перестановками чисел $1, \dots, N$. Вычислите количество возможных различных массивов, которые могут получиться у Snuke, и выведите ответ по модулю $10^9 + 7$.

Input

Первая строка входа содержит одно целое число N ($1 \leq N \leq 2000$). Вторая строка содержит N целых чисел P_i ($1 \leq P_i \leq N$, $P_i \neq P_j$, если $i \neq j$). Третья строка содержит N целых чисел Q_i ($1 \leq Q_i \leq N$, $Q_i \neq Q_j$, если $i \neq j$).

Output

Выведите ответ к задаче.

Examples

standard input	standard output
4 3 1 2 4 3 1 2 4	14
10 5 7 3 1 6 4 2 10 9 8 2 8 9 1 5 6 10 4 3 7	127224

Problem E. Mirror Rice Cake

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

Зеркальный рисовый пирог (стопка рисовых пирогов) — японское кушанье, которое готовится на Новый Год.

У Snuke есть N рисовых пирогов и он хочет сложить их в стопку.

Вес i -го рисового пирога равен a_i . Чтобы собрать Зеркальный рисовый пирог, нужно сложить пироги один на другого так, чтобы для каждого рисового пирога суммарный вес всех пирогов выше него был строго меньше, чем его собственный вес.

Snuke хочет узнать, какое наибольшее количество рисовых пирогов он может использовать для Зеркального рисового пирога.

Input

Первая строка входа содержит одно целое число N ($1 \leq N \leq 1000$). Каждая из последующих N строк содержит вес a_i очередного пирога ($1 \leq a_i \leq 10^9$).

Output

Выведите одно целое число — ответ к задаче.

Example

standard input	standard output
5	3
3	
20	
5	
8	
6	

Note

В примере к задаче пироги можно расположить 3, 5, 20 сверху вниз.

Problem F. Number Cards

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

У Snuke N карточек с цифрами. i -я карточка содержит целое положительное число a_i и окрашена в цвет с номером c_i .

Snuke сделал следующее предположение о возможной раскраске этих карточек:

- Карточки с номерами $1 \leq a_i \leq M$ окрашены в один и тот же цвет.
- Карточки с номерами $M+1 \leq a_i \leq 2M$ окрашены в один и тот же цвет, не совпадающий с цветом ни одной карточки с номерами $1 \leq a_i \leq M$.
- Карточки с номерами $2M+1 \leq a_i \leq 3M$ окрашены в один и тот же цвет, не совпадающий с цветом ни одной карточки с номерами $1 \leq a_i \leq 2M$.
- Карточки с номерами $3M+1 \leq a_i \leq 4M$ окрашены в один и тот же цвет, не совпадающий с цветом ни одной карточки с номерами $1 \leq a_i \leq 3M$.
- и так далее

Сколько целых положительных чисел M удовлетворяют этому предположению?

Input

Первая строка входа содержит одно целое число N ($1 \leq N \leq 20$). Каждая из последующих N строк содержит два целых числа a_i и c_i — число, написанное на очередной карте, и номер её цвета, соответственно ($1 \leq a_i \leq 10^9$, $1 \leq c_i \leq 20$).

Гарантируется, что последовательность a_i является строго возрастающей.

Output

Выведите ответ к задаче. Если количество соответствующих M бесконечно, выведите -1 .

Examples

standard input	standard output
4 27 2 2000 4 2015 4 2100 1	277
3 1 1 2 2 3 1	0

Problem H. Random Walk

Input file: *standard input*
Output file: *standard output*
Time limit: 3.5 seconds
Memory limit: 512 mebibytes

Дан бесконечный лист клетчатой бумаги. Snuke собирается заняться случайным блужданием на этом листе следующим образом:

Он начинает движение из точки $(0, 0)$ и делает N шагов. Когда он находится в (i, j) , его позиция после следующего шага — одна из точек $(i - 1, j)$, $(i, j - 1)$, $(i, j + 1)$, и $(i + 1, j)$. Все варианты равновероятны и их вероятность равна $\frac{1}{4}$.

Пусть E — математическое ожидание количества посещённых точек. Вычислите значение $E \times 4^N$ по модулю M (гарантируется, что это число будет целым). Заметим, что $(0, 0)$ всегда считается посещённой.

Input

Вход состоит из двух целых чисел N и M ($1 \leq N \leq 5000$, $10^9 \leq M \leq 2 \times 10^9$).

Output

Выведите ответ в отдельной строке.

Examples

standard input	standard output
2 1000000007	44
2015 2000000000	1892319232

Problem I. Robots

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

У Snuke N роботов. Они пронумерованы числами от 1 до N .

Первоначально робот i находится в точке (x_i, y_i) и повернут в сторону d_i (d_i — один из символов 'U', 'D', 'L' и 'R' — движение в направлении возрастания y , убывания y , убывания x и возрастания x , соответственно).

Изначально все роботы неподвижны. Когда до робота кто-то дотрагивается (робот или Snuke), он немедленно начинает движение в сторону, в которую он повернут, с единичной скоростью. При этом движущиеся роботы умеют минимизировать последствия столкновений (то есть если робот уже движется, контакты с другими роботами не влияют на его траекторию).

Snuke дотронулся до робота с номером 1 в момент времени 0. Вычислите координаты каждого робота в момент времени T .

Input

Первая строка входа содержит два целых числа N и T ($1 \leq N \leq 10^5$, $0 \leq T \leq 10^{18}$). i -я из последующих N строк содержит два целых числа x_i и y_i и символ d_i — первоначальные координаты и направление движения i -го робота ($0 \leq x_i, y_i \leq 10^9$, d_i — один из символов 'U', 'D', 'L', 'R'). В момент времени 0 никакие два робота не находятся в одной и той же точке.

Output

Выведите N строк. В i -й из них выведите координаты i -го робота в момент времени T .

Example

standard input	standard output
5 10	1 10
1 0 U	3 6
3 1 U	9 2
1 2 R	-8 1
1 1 L	8 1
0 1 R	

Problem J. Ropes

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

N человек спят в комнате. Они пронумерованы последовательными целыми числами от 1 до N . Snuke решил подшутить над ними и связать друг с другом, используя $N - 1$ верёвку, при этом:

- Концы каждой верёвки должны быть привязаны к разным людям;
- Цепочка верёвок должны быть связной (то есть от каждого человека до любого другого можно пройти по одной или более верёвкам).
- К i -му человеку должно быть привязано ровно a_i верёвок.

Вычислите количество различных способов связывания, удовлетворяющих вышеприведённым условиям. Способы считаются различными, если существует как минимум два участника, который связаны напрямую верёвкой в одном случае и не связаны в другом.

Input

Первая строка входа содержит одно целое число N ($2 \leq N \leq 10^5$). i -я из последующих N строк содержит целое число a_i — количество верёвок, которые должны быть привязаны к i -му человеку ($1 \leq a_i \leq 3$).

Output

Выведите одно число — ответ к задаче.

Example

standard input	standard output
9	1260
1	
3	
2	
1	
3	
1	
2	
1	
2	

Problem L. String Modification

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

На Новый Год Snuke получил от Укконена в подарок строку s . Определите, сможет ли Snuke переделать эту строку в строку t , повторяя какое-то (возможно, нулевое) количество раз следующую операцию:

Выбрать символ в строке s и вставить после этого символа какой-либо другой, обязательно отличный от данного.

Например, он может переделать “abca” в “adbca” за одну операцию, выбрав первую букву ‘a’ и вставив сразу после неё ‘d’. А вот “abca” в “aabca” он переделать не может.

Input

Первая строка входа содержит строку s , вторая — строку t . Обе строки составлены из строчных латинских букв и для них выполняется ограничение $1 \leq |s| \leq |t| \leq 5000$.

Output

Выведите “Yes” в случае, когда Snuke может переделать s в t , и “No” в противном случае.

Examples

standard input	standard output
snuke snukent	Yes
snuke ssnuke	No

Problem N. Soccer Match

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

Результатом футбольного матча для одной команды может быть либо победа, либо ничья, либо поражение. За победу команда получает 3 очка, за ничью — одно очко, за поражение — 0.

В старой книге по истории футбола вы нашли упоминание о том, что в первом сезоне в профессиональной лиге команда, за которую Вы болеете, сыграла N матчей и набрала K очков.

Вы хотите выяснить список всех возможных распределений результатов (то есть количество побед, ничьих и поражений в сезоне). Напишите программу, которая выводит все такие распределения.

Input

Вход содержит два целых числа N ($0 < N \leq 100$) и K ($0 \leq K \leq 300$) — количество сыгранных матчей и количество набранных при этом очков, разделённые пробелом.

Гарантируется, что существует как минимум одно возможное сочетание побед, ничьих и поражений, при которых сыграно N матчей и набрано K очков.

Output

Выведите список возможных результатов по одному в строке. Каждый результат состоит из трёх целых чисел, разделённых пробелами — количества побед w , количества ничьих t и количества поражений l . Результаты должны выводиться в порядке убывания количества побед.

Example

standard input	standard output
6 10	3 1 2 2 4 0