Задача A. Колония бактерий

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Ученые обнаружили новый вид бактерии и начали проводить эксперименты для его изучения.

В одном из экспериментов они поместили колонию бактерий на бесконечное клетчатое поле, и оказалось, что каждую секунду она расширяется. В каждую чётную секунду колония расширяется в восьми направлениях, то есть занимает клетки, соседние с занятыми по стороне и диагонали, если они ещё не заняты, а в нечётную – только в четырёх направлениях, в клетки, соседние с занятыми по стороне.

Помогите учёным определить, сколько клеток занимает колония бактерий в k-ю секунду эксперимента, если она была помещена на клетчатое поле в первую секунду.

Формат входных данных

В первой строке дано целое число k — время в секундах, когда учёные хотят понять, сколько клеток занимает колония бактерий ($1 \leqslant k \leqslant 10^8$).

Формат выходных данных

Вывести одно число – количество занятых колонией бактерий клеток в k-ю секунду эксперимента.

Примеры

стандартный ввод	стандартный вывод
1	1
2	9
3	21
4	45
5	69

Замечание

Заполнение поля первые пять секунд, в клетке указана секунда, когда впервые эта клетка будет занята колонией бактерий.

		5	5	5	5	5		
	5	4	4	4	4	4	5	
5	4	4	3	3	3	4	4	5
5	4	3	2	2	2	3	4	5
5	4	3	2	1	2	3	4	5
5	4	3	2	2	2	3	4	5
5	4	4	3	3	3	4	4	5
	5	4	4	4	4	4	5	
		5	5	5	5	5		

Задача B. Двухэтажный адвент-календарь

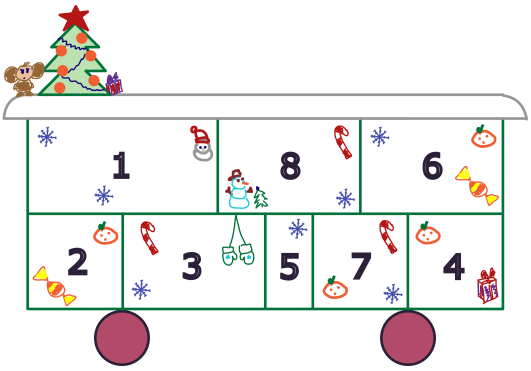
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

«Генеральная пассажирская компания» запускает новогодние железнодорожные экскурсии из Санкт-Петербурга в Великий Устюг. Для всех покупателей этой поездки разработали особенный подарок — адвент-календарь.

Адвент-календарь представляет собой коробку в виде главного перевозчика «Генеральной пассажирской компании» — двухэтажного вагона. В коробке в два уровня лежат коробочки, в каждой коробочке по конфете. На верхнем уровне находится n коробочек, на нижнем уровне находится m коробочек. На каждой коробочке написано натуральное число от 1 до $n + m$ включительно. Числа на коробочках не повторяются.

Для каждой коробочки известна её длина. Коробочки могут отличаться по длине. Гарантируется, что суммы длин коробочек на первом и втором уровнях адвент-календаря совпадают.

Чтобы правильно открыть адвент-календарь, нужно в первый день вытащить и открыть коробочку с номером 1, во второй день — коробочку с номером 2 и так далее, завершает календарь коробочка с номером $n + m$, которую следует вытащить и открыть в $(n + m)$ по счёту день. Пример адвент-календаря на рисунке.



Адвент-календарь на 8 ячеек. Чтобы правильно открыть его и создать новогоднее настроение, за 8 дней до Нового года нужно открыть ячейку 1, за 7 дней ячейку 2 и так далее. В последний день — 31 декабря — нужно открыть ячейку 8.

Дизайнер и перфекционист Майя решила прокатиться на Новый год в поезде и получила в подарок двухэтажный адвент-календарь. Майя считает неудобным, когда она открывает коробочку с конфетой на нижнем уровне, а сверху на ней на верхнем уровне лежит хотя бы одна ещё не открытая коробочка.

Майе стало интересно, сколько коробочек заранее нужно вытащить из календаря, чтобы он стал удобным. При этом Майе хочется, чтобы в адвенте осталось как можно больше коробочек. Помогите ей ответить, какое минимальное количество коробочек нужно заранее вытащить из календаря, чтобы при открытии коробочки на нижнем уровне на ней не лежала ни одна ещё не открытая коробочка из верхнего уровня. Заранее коробочки можно вытаскивать как с верхнего, так и с нижнего уровня.

Формат входных данных

На первой строке ввода находится целое число n ($1 \leqslant n \leqslant 10^5$) — количество коробочек на верхнем уровне календаря.

В следующих n строках находятся два числа a_i и x_i ($1 \leqslant a_i \leqslant 10^9$, $1 \leqslant x_i \leqslant n + m$) — длина i-й слева коробочки на верхнем уровне календаря и номер, который на ней написан, соответственно.

На $(n + 1)$ строке ввода находится целое число m ($1 \leqslant m \leqslant 10^5$) — количество коробочек на втором уровне календаря.

В следующих m строках находятся два числа b_j и y_j ($1 \leqslant b_j \leqslant 10^9$, $1 \leqslant y_j \leqslant n + m$) — длина j-й слева коробочки на нижнем уровне календаря и номер, который на ней написан, соответственно.

Гарантируется, что $a_1+a_2+\ldots+a_n = b_1+b_2+\ldots+b_m$. Гарантируется, что все числа в множестве $\{x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m\}$ различны.

Формат выходных данных

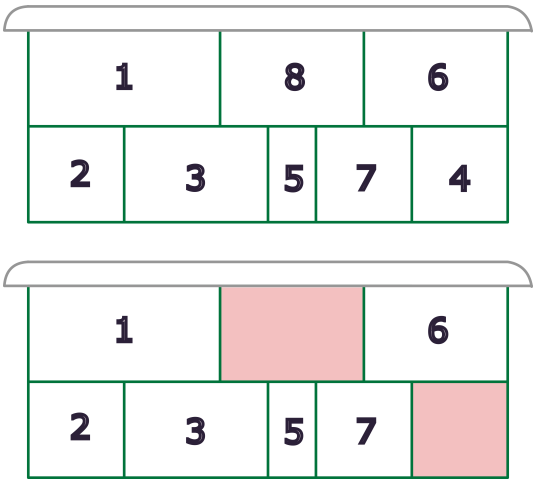
Выведите единственное число k — минимальное количество коробочек, которые надо достать из календаря, чтобы он стал удобным.

Примеры

стандартный ввод	стандартный вывод
3 1 1 1 2 1 3 3 1 4 1 5 1 6	0
3 4 1 3 8 3 6 5 2 2 3 3 1 5 2 7 2 4	2

Замечание

Во втором примере можно убрать коробки с номерами 4 и 8. После этого календарь будет выглядеть как на рисунке ниже и станет удобным.



Задача C. Промежуточная вертикальность

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

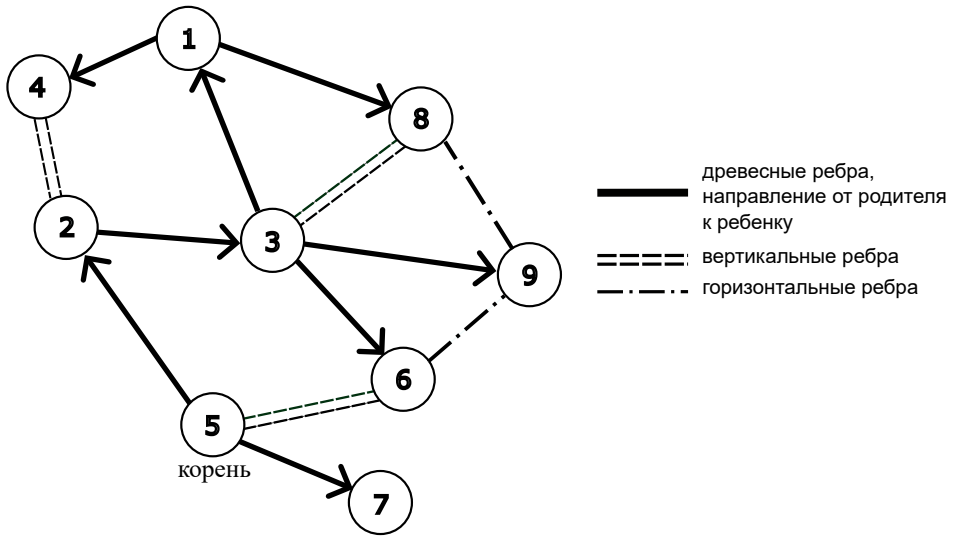
Два классических алгоритма на графах — обход в глубину и обход в ширину — строят в графе два остовных дерева. При этом обход в глубину известен тем, что в полученном дереве нет *горизонтальных* рёбер — рёбер, соединяющих вершины, которые не являются предками друг друга, а обход в ширину — тем, что в полученном дереве нет *вертикальных* рёбер — рёбер, соединяющих вершину с предком в дереве. В этой задаче вам понадобится построить промежуточное остовное дерево, имеющее заданное число горизонтальных и вертикальных рёбер.

Напомним, что неориентированный граф состоит из множества вершин V и множества рёбер E, где каждое ребро соединяет две вершины. Будем рассматривать связные графы, то есть такие графы, в которых можно от любой вершины добраться до любой другой по рёбрам. Дерево — это связный неориентированный граф, не содержащий циклов, а остовное дерево в графе — подмножество его рёбер, которое образует дерево, по которому можно добраться от любой вершины графа до любой другой. Напомним два основных свойства дерева: в дереве с n вершинами ровно $n-1$ рёбер, а между любыми двумя вершинами в дереве есть ровно один путь.

Выделим в графе вершину r, которую будем называть *корнем* дерева. Вершины, которые лежат на единственном пути от вершины x до вершины r, называются *предками* вершины x, а первая вершина на этом пути называется *родителем* вершины x и обозначается как p_x. У корня родителя нет.

Если в графе зафиксирован корень и остовное дерево, то все рёбра графа можно разделить на три типа:

- *древесные* — рёбра выбранного остовного дерева;
- *вертикальные* — рёбра, не принадлежащие дереву, соединяющие вершину с её предком;
- *горизонтальные* — остальные рёбра графа.



Вертикальностью остовного дерева в графе будем называть количество вертикальных рёбер.

Вам задан граф с n вершинами и m рёбрами, корень дерева r и число h, $0 \leqslant h \leqslant m - n + 1$. Необходимо построить остовное дерево заданного графа с корнем в r, имеющее вертикальность, равную h, либо сообщить, что такого дерева не существует.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит одно целое число t — количество наборов входных данных ($1 \leqslant t \leqslant 10^5$). Далее следуют одно за другим описания t наборов входных данных.

В первой строке каждого набора входных данных находится четыре целых числа n, m, r и h — количество вершин и рёбер в графе, соответственно, номер корневой вершины и требуемая вертикальность будущего остовного дерева ($2 \leqslant n \leqslant 3 \cdot 10^5$; $n - 1 \leqslant m \leqslant 3 \cdot 10^5$; $1 \leqslant r \leqslant n$; $0 \leqslant h \leqslant m - n + 1$).

В каждой из следующих m строк находится по два целых числа u_i, v_i — номера вершин, соединённых ребром в графе ($1 \leqslant u_i, v_i \leqslant n$; $u_i \neq v_i$).

Гарантируется, что все графы связные, не содержат петель и кратных рёбер. Гарантируется, что сумма n по всем наборам входных данных не превосходит $3 \cdot 10^5$. Гарантируется, что сумма m по всем наборам входных данных не превосходит $3 \cdot 10^5$.

Формат выходных данных

Для каждого тестового случая найдите требуемое остовное дерево T и выведите в отдельной строке n целых чисел $p_1, p_2, \ldots, p_n$, где p_i — номер родителя i-й вершины в дереве T ($1 \leqslant p_i \leqslant n$). В качестве p_r можете вывести любое число от 1 до n. Если дерева T, обладающего нужными свойствами, не существует, выведите вместо этого n чисел -1.

Пример

стандартный ввод	стандартный вывод
4 5 7 2 0 1 2 2 3 3 4 4 1 3 5 5 4 1 5 5 7 2 1 1 2 2 3 3 4 4 1 3 5 5 4 1 5 5 7 2 2 1 2 2 3 3 4 4 1 3 5 5 4 1 5 5 7 2 3 1 2 2 3 3 4 4 1 3 5 5 4 1 5	2 1 2 3 3 2 1 4 1 1 2 1 5 5 1 2 1 4 1 3

Задача D. Два массива

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Обозначим максимум в массиве d как $\max(d)$, а минимум как $\min(d)$.

Заданы два массива a и b длины n. За одно действие можно выбрать индекс $1 \leqslant i \leqslant n$ и одновременно увеличить элементы a_i и b_i на единицу: $a_i = a_i + 1$, $b_i = b_i + 1$. Необходимо с использованием этих действий добиться того, чтобы одновременно выполнились два условия:

- $\max(a) - \min(a) \leqslant x$,
- $\max(b) - \min(b) \leqslant y$.

Определите минимальное количество действий, чтобы добиться одновременного выполнения указанных условий, или выясните, что это невозможно.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t — количество наборов входных данных ($1 \leqslant t \leqslant 10^5$). Далее следует описание наборов входных данных.

Первая строка каждого набора данных содержит три целых числа: n, x, y ($1 \leqslant n \leqslant 10^5$, $0 \leqslant x, y \leqslant 10^9$).

Во второй строке каждого набора заданы n целых чисел $a_1, a_2, \ldots a_n$ — элементы массива a ($-10^9 \leqslant a_i \leqslant 10^9$).

В третьей строке каждого набора заданы n целых чисел $b_1, b_2, \ldots b_n$ — элементы массива b ($-10^9 \leqslant b_i \leqslant 10^9$).

Гарантируется, что сумма n по всем наборам входных данных не превосходит 10^5.

Формат выходных данных

Для каждого набора входных данных выведите одно целое число — минимально возможное количество действий, необходимых для выполнения обоих условий. Если оба условия одновременно выполнить невозможно, выведите -1.

Пример

стандартный ввод	стандартный вывод
5 4 2 3 -1 -2 -3 -4 -1 -2 -3 -4 3 3 2 1 6 4 1 4 1 4 0 3 0 2 1 2 0 2 3 3 5 2 1 -1 0 1 2 3 2 2 2 2 2 3 66 77 235 -111 9 100 -200 -100	1 3 3 -1 440

Задача E. По классике

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Вероятно, вам знакома классическая задача нахождения наибольшей возрастающей подпоследовательности в массиве. Пусть a — массив, состоящий из n целых чисел. Назовем подпоследовательность $i_1 < i_2 < \ldots < i_k$ *возрастающей*, если $a_{i_1} < a_{i_2} < \ldots < a_{i_k}$. Наибольшей возрастающей подпоследовательностью называется возрастающая подпоследовательность максимальной длины. Конечно, мы не будем предлагать вам решить классическую задачу, вам предстоит решить ее усложненную версию...

Изначально есть пустой массив a. Далее в массив последовательно добавляются числа $1, 2, \ldots, n$ в этом порядке. При этом число i добавляется в массив на позицию p_i. Позиции в массиве нумеруются целыми числами от 1 до k, где k — текущий размер массива. При добавлении элемента на позицию p в массив размера k все элементы, которые раньше имели позиции от p до k, сдвигаются на одну позицию вправо, а на освободившееся место добавляется текущий элемент.

Ваша задача — определить длину наибольшей возрастающей подпоследовательности в массиве после каждого добавления нового элемента.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leqslant n \leqslant 200\,000$) — количество добавленных элементов.

Вторая строка содержит n целых чисел $p_1, p_2, \ldots, p_n$ ($1 \leqslant p_i \leqslant i$) — p_i обозначает позицию, на которую добавляется элемент i.

Формат выходных данных

Выведите n целых чисел — размер наибольшей возрастающей подпоследовательности массива после каждого добавления нового элемента.

Примеры

стандартный ввод	стандартный вывод
5 1 2 1 3 4	1 2 2 2 3
1 1	1

Замечание

Массив в первом примере изменялся следующим образом: $[] \to [1] \to [1, 2] \to [3, 1, 2] \to [3, 1, 4, 2] \to [3, 1, 4, 5, 2]$.

Задача F. Обмен и удаление

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Есть классический способ удаления элемента из массива: поменять местами значения удаляемого элемента и последнего элемента массива, после чего удалить последний элемент. К сожалению, оказалось, что такой способ не всегда сохраняет порядок элементов массива. Ваша задача — посчитать количество последовательностей из k удалений, после которых изначально упорядоченный по возрастанию массив останется упорядоченным.

Дан массив a длины n, исходно заполненный числами от 1 до n по возрастанию, то есть $a_i = i$, а также массив b длины k, элементы которого — попарно различные числа от 1 до n.

Выполняется k шагов, на j-м шаге происходит следующее: выбирается такое i от 1 до $n-j+1$, что $a_i = b_j$, после чего a_i и a_{n-j+1} обмениваются местами (если $i = n - j + 1$, то ничего не происходит). Затем последний на данном шаге элемент массива — он имеет номер $n-j+1$ — удаляется из массива.

Массив $[b_1, b_2, \ldots, b_k]$ будем называть *хорошим*, если после выполнения k шагов массив $[a_1, a_2, \ldots, a_{n-k}]$ является строго возрастающим.

Вам даны числа n и k, посчитайте количество хороших массивов $[b_1, b_2, \ldots, b_k]$. Ответ может оказаться слишком большим, поэтому выведите остаток от деления ответа на $10^9 + 7$.

Формат входных данных

В первой строке задано целое число t ($1 \leqslant t \leqslant 10^4$) — количество наборов входных данных. В следующих t строках заданы наборы входных данных.

В первой строке набора входных данных заданы целые числа n и k ($1 \leqslant n \leqslant 5 \cdot 10^5$, $0 \leqslant k \leqslant n$).

Гарантируется, что сумма n по всем наборам входных данных не превышает $5 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных выведите одно целое число — количество хороших массивов b, взятое по модулю $10^9 + 7$.

Пример

стандартный ввод	стандартный вывод
5 1 0 1 1 2 2 3 1 4 2	1 1 2 2 7

Замечание

Разберем четвертый тест из примера. В нем $n = 3$ и $k = 1$. Изначально $a = [1, 2, 3]$. Посмотрим, как изменится a после первого шага для всех возможных массивов b.

- $b = [1]$. Тогда a меняется следующим образом: $[1, 2, 3] \to [3, 2, 1] \to [3, 2]$. Массив $[3, 2]$ не является возрастающим.
- $b = [2]$. Тогда a меняется следующим образом: $[1, 2, 3] \to [1, 3, 2] \to [1, 3]$. Массив $[1, 3]$ является возрастающим.
- $b = [3]$. Тогда a меняется следующим образом: $[1, 2, 3] \to [1, 2, 3] \to [1, 2]$. Массив $[1, 2]$ является возрастающим.

Получаем, что существует два хороших массива $b = [2]$ и $b = [3]$, поэтому ответ на четвертый тест из примера равен 2.

Задача G. Трасса М-11

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Новая скоростная автомобильная магистраль М-11 представляет собой бесконечную прямую.

На трассе расположены n остановочных пунктов, каждый из которых — зона отдыха или заправка. Каждый остановочный пункт задается своей координатой x_i и при этом никакие два остановочных пункта не находятся в одном и том же месте. Тройка остановочных пунктов (i, j, k) называется удобной, если $x_i < x_j < x_k$, в точках x_i и x_k расположены заправки, а в точке x_j — зона отдыха, и при этом расстояние между заправками не превосходит d.

Одна команда из Москвы собирается поехать на ВКОШП по трассе М-11, и её руководителю стало интересно, сколько существует удобных троек остановок на пути.

Формат входных данных

В первой строке дано два натуральных числа n и d — количество остановочных пунктов и максимальное расстояние между заправками ($3 \leqslant n \leqslant 5 \cdot 10^5$, $2 \leqslant d \leqslant 10^9$).

В следующих n строках даны остановочные пункты. Каждый остановочный пункт задается двумя целыми числами x_i и t_i — координата пункта и его тип. Тип 0 обозначает зону отдыха, тип 1 обозначает заправку ($-10^{18} \leqslant x_i \leqslant 10^{18}$; $t_i \in \{0, 1\}$). Гарантируется, что координаты остановочных пунктов идут в возрастающем порядке.

Формат выходных данных

Выведите единственное число — количество удобных троек.

Примеры

стандартный ввод	стандартный вывод
8 5 1 1 2 0 3 1 6 0 7 0 8 1 15 1 19 1	3
10 6 0 1 1 0 3 1 4 0 5 1 8 1 10 0 11 0 14 1 18 1	7

Замечание

В первом наборе входных данных удобными тройками являются $(1, 2, 3)$, $(3, 4, 6)$ и $(3, 5, 6)$.

Задача H. Роботы-исследователи

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Поле для испытания роботов состоит из n полей, пронумерованных от 1 до n слева направо. На каждом поле написана буква английского алфавита, при прочтении букв на полях от первого до n-го получается строка s.

Есть два робота, которые могут передвигаться по полю, при этом:

- роботам известна строка s;
- роботы могут свободно обмениваться информацией;
- роботы всегда знают расстояние между ними, а также какой из роботов находится левее, а какой правее;
- каждый из двух роботов может прочитать букву, которая находится под ним;

За один шаг робот может, обменявшись информацией с другим роботом, переместиться на соседнее поле слева или на соседнее поле справа. При этом, если робот пытается переместиться левее первого или правее n-го поля, то он уничтожается.

Ученые хотят провести q экспериментов, в i-м из которых первого робота ставят на позицию x_i, а второго на позицию y_i. Цель роботов в каждом из экспериментов — посетить как можно больше различных полей. Для каждого эксперимента необходимо определить, какое максимальное количество полей смогут посетить роботы, не рискуя уничтожением.

Формат входных данных

В первой строке указана пара чисел n и q ($1 \leqslant n, q \leqslant 300\,000$) — количество полей и количество экспериментов.

Во второй строке указана строка s длины n, состоящая из n строчных букв латинского алфавита.

В последующих q строках указаны пары чисел x_i, y_i ($1 \leqslant x_i, y_i \leqslant n$).

Формат выходных данных

Для каждого из экспериментов выведите наибольшее количество различных полей, которое смогут посетить роботы.

Пример

стандартный ввод	стандартный вывод
10 4 aabaabbaab 4 5 8 5 2 3 1 1	3 10 3 3

Замечание

Рассмотрим последний эксперимент в примере. Роботы стартуют в одной точке и видят букву «a». Они понимают, что двигаться влево опасно, это может привести к уничтожению робота. Однако двигаться направо безопасно, так как последняя буква строки — «b». Один или оба робота двигаются вправо, пока не окажутся на букве «b». Оказавшись на букве «b», роботы понимают, что только что считали строку «aab», эта строка может быть как началом, так и концом строки, выйти за ее пределы, не рискуя уничтожением, они не могут, поэтому всего удалось посетить всего 3 поля.

Задача I. Шалость

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Катя составила слово s_1 из кубиков, но когда она вернулась в комнату, увидела, как оттуда выбегал её брат Андрей. Теперь слово из кубиков выглядело иначе — s_2. Андрей признался, что он несколько раз сделал *шалость*. Его шалость заключалась в следующем. Андрей выбирал позицию, потом в неё вставлял рядом два кубика, на которых написана одинаковая буква. Он мог поставить эти два кубика в начале строки, в конце строки или между двумя соседними кубиками.

Помогите Кате определить, сказал ли Андрей правду, то есть могла ли получиться строка s_2 из строки s_1 путём применения, возможно, нескольких *шалостей*.

Формат входных данных

Один тест содержит несколько наборов входных данных.

В первой строке дано одно целое число t — количество наборов входных данных ($1 \leqslant t \leqslant 500\,000$).

В первой строке каждого описания набора дана одна строка s_1 — слово из кубиков, которое было у Кати изначально.

Во второй строке каждого описания набора дана одна строка s_2 — слово из кубиков, которое получил Андрей.

Гарантируется, что все слова состоят из строчных латинских букв. Суммарная длина всех слов не превышает $1\,000\,000$.

Формат выходных данных

Для каждого набора входных данных выведите «YES», если Андрей мог сказать правду, и «NO» в противном случае.

Пример

стандартный ввод	стандартный вывод
2 hello havvaeeello test tesssst	YES NO

Задача J. Кошмарная сумма

Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Дан массив a длины n, состоящий из различных положительных целых чисел. Вычислите

$$\sum_{l=1}^{n}\sum_{r=l}^{n}\left\lfloor\frac{\max(a_l, a_{l+1}, \ldots, a_r)}{\min(a_l, a_{l+1}, \ldots, a_r)}\right\rfloor$$

Здесь $\lfloor x\rfloor$ означает x, округленный вниз до ближайшего целого.

Таким образом, необходимо вычислить сумму результатов целочисленного деления максимума на минимум по всем подотрезкам массива a.

Формат входных данных

В первой строке ввода находится единственное целое число n — длина массива ($1 \leqslant n \leqslant 300\,000$).

Во второй строке ввода находятся n целых чисел — массив a ($1 \leqslant a_i \leqslant 300\,000$).

Гарантируется, что все числа в массиве a различны.

Формат выходных данных

Выведите единственное число — искомую сумму.

Пример

стандартный ввод	стандартный вывод
6 1 3 6 4 2 5	56

Замечание

Рассмотрим пример подробнее:

$[l, r]$	$a[l \ldots r]$	min	max	$\left\lfloor\frac{\max}{\min}\right\rfloor$
$[1, 1]$	$[1]$	1	1	1
$[1, 2]$	$[1, 3]$	1	3	3
$[1, 3]$	$[1, 3, 6]$	1	6	6
$[1, 4]$	$[1, 3, 6, 4]$	1	6	6
$[1, 5]$	$[1, 3, 6, 4, 2]$	1	6	6
$[1, 6]$	$[1, 3, 6, 4, 2, 5]$	1	6	6
$[2, 2]$	$[3]$	3	3	1
$[2, 3]$	$[3, 6]$	3	6	2
$[2, 4]$	$[3, 6, 4]$	3	6	2
$[2, 5]$	$[3, 6, 4, 2]$	2	6	3
$[2, 6]$	$[3, 6, 4, 2, 5]$	2	6	3
$[3, 3]$	$[6]$	6	6	1
$[3, 4]$	$[6, 4]$	4	6	1
$[3, 5]$	$[6, 4, 2]$	2	6	3
$[3, 6]$	$[6, 4, 2, 5]$	2	6	3
$[4, 4]$	$[4]$	4	4	1
$[4, 5]$	$[4, 2]$	2	4	2
$[4, 6]$	$[4, 2, 5]$	2	5	2
$[5, 5]$	$[2]$	2	2	1
$[5, 6]$	$[2, 5]$	2	5	2
$[6, 6]$	$[5]$	5	5	1

Задача K. Криптография Пети

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Петя получил двойку по криптографии, но это преподаватели не поняли гения, а не он лентяй. Чтобы убедить всё мировое сообщество в своей гениальности, Петя создал новую систему шифрования с открытым ключом — PSA. У любой уважающей себя криптосистемы с открытым ключом есть открытый и закрытый ключ.

В качестве закрытого ключа Петя выбрал дерево T, а в качестве открытого — два числа (n, p), где n равно числу вершин в дереве T, а p — количество путей длины 2 в T. Напомним, что дерево представляет собой неориентированный связный граф, не содержащий циклов.

Особенность Петиной криптосистемы в том, что для её взлома подойдет любой закрытый ключ, которому соответствует открытый. Но ведь это не проблема, Петя выбрал довольно сложную задачу, верно? Восстановите любой закрытый ключ криптосистемы PSA или скажите, что такого открытого ключа не могло получиться.

Формат входных данных

В единственной строке ввода содержится два числа n и p ($1 \leqslant n \leqslant 1000$, $0 \leqslant p \leqslant 10^9$).

Формат выходных данных

Если ответ существует, в первой строке выведите «`Yes`». В следующих $n - 1$ строке выведите по два различных целых числа от 1 до n — рёбра дерева.

Если ответа не существует, то в единственной строке выведите «`No`».

Примеры

стандартный ввод	стандартный вывод
7 11	Yes 1 2 2 3 3 4 3 5 3 6 3 7
5 5	No

Задача L. Два самоката

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Катя знает, что время поездки от дома до метро на самокате составляет t секунд.

Стоимость поездки на самокате компании `W` вычисляется следующим образом: сначала определяется количество полных минут, затраченных на дорогу, а затем это число умножается на 60 и используется для вычисления стоимости из расчета c_1 копеек за секунду.

Стоимость поездки на самокате компании `Y` рассчитывается иначе: сначала вычисляется стоимость поездки по тарифу c_2 копеек за секунду, а затем эта сумма округляется до целых рублей вверх.

Помогите Кате понять, за какую наименьшую стоимость она может добраться до метро на самокате одной из этих компаний.

Напомним, что в одном рубле содержится 100 копеек.

Формат входных данных

В первой строке даны три целых числа t, c_1 и c_2 — время поездки в секундах, тариф поездки в копейках в секунду на самокате компании `W` и тариф поездки в копейках в секунду на самокате компании `Y` ($1 \leqslant t \leqslant 1000$, $10 \leqslant c_1, c_2 \leqslant 20$).

Формат выходных данных

Выведите минимальную стоимость поездки в копейках.

Пример

стандартный ввод	стандартный вывод
473 10 11	4200