Задача A. Обратный отсчет

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

У Дениса есть массив из n положительных целых чисел, i-е целое число массива равно a_i.

Подотрезок массива называется m-отсчетом, если он имеет длину m и содержит целые числа $m, m-1, m-2, ..., 2, 1$ в указанном порядке. Например, $[3, 2, 1]$ — это 3-отсчет.

Сколько k-отсчетов в массиве Дениса?

Формат входных данных

Первая строка содержит два целых числа n и k ($2 \leqslant k \leqslant n \leqslant 2 \cdot 10^5$). Вторая строка содержит n целых чисел a_i ($1 \leqslant a_i \leqslant 2 \cdot 10^5$).

Формат выходных данных

Выведите одно число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
13 3 1 2 3 7 9 3 2 1 8 3 2 1 1	2
4 2 101 100 99 98	0
8 6 10 7 6 5 4 3 2 1	1

Задача B. Разкраченность последовательности

Ограничение по времени: 0.7 секунд
Ограничение по памяти: 512 мегабайт

Даша пишет генератор случайных чисел. Она хорошо понимает, что в итоге получит лишь псевдослучайный генератор, поэтому решает использовать максимально простой способ генерации: начиная с некоторого числа x_0, она строит последовательность по рекуррентной формуле:

$$x_{i+1} = (a \cdot x_i + b) \bmod c, \quad i \geqslant 0,$$

где a, b, c и x_0 — заранее заданные положительные целые числа.

Таким образом, Даша получает последовательность из n чисел:

$$x_0, x_1, x_2, \ldots, x_{n-1}.$$

Чтобы оценить качество полученной последовательности, Даша вводит величину, которую называет разкраченностью. Для её вычисления необходимо отсортировать элементы последовательности по возрастанию и найти максимальную разность между двумя соседними элементами в отсортированном порядке. Формально:

$$\text{разкраченность} = \max_{1 \leqslant i < n} (y_i - y_{i-1}),$$

где $y_0 \leqslant y_1 \leqslant \ldots \leqslant y_{n-1}$ — отсортированная последовательность $\{x_i\}$.

Даша сгенерировала несколько последовательностей по описанному методу и хочет вычислить разкраченность каждой из них. Помогите ей!

Формат входных данных

В первой строке входных данных содержится единственное целое число t $(1 \leqslant t \leqslant 10)$ — количество последовательностей, которые сгенерировала Даша. Далее следует их описание, каждое из которых состоит из двух строк.

В первой строке содержится одно целое число n $(2 \leqslant n \leqslant 5 \cdot 10^6)$ — длина последовательности.

Во второй строке содержится четыре целых числа a, b, c и x_0 $(0 \leqslant a, b \leqslant 10^9,\ 1 \leqslant c \leqslant 10^9,\ 0 \leqslant x_0 < c)$.

Формат выходных данных

Для каждой последовательности выведите в отдельной строке одно целое число — её разкраченность.

Пример

стандартный ввод	стандартный вывод
1 7 2 3 11 4	4

Замечание

Разберём тест из условия задачи:

- Последовательность, которую генерирует псевдослучайный генератор Даши, выглядит так: $x = [4, 0, 3, 9, 10, 1, 5]$.
- Расположив эти числа в отсортированном порядке, мы получаем $y = [0, 1, 3, 4, 5, 9, 10]$.
- Разкраченность этой последовательности равна $\max(1 - 0, 3 - 1, 4 - 3, 5 - 4, 9 - 5, 10 - 9) = 4$.

Задача C. Прыжки по кругу

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вам даны целое число n и две перестановки целых чисел от 1 до n: $s_1, s_2, \ldots, s_n$ и $t_1, t_2, \ldots, t_n$.

Обе перестановки задают числа, стоящие по кругу, поэтому мы считаем, что $s_1 = 1$ и $t_1 = 1$. Более того, мы считаем, что перестановки s и t равны, если существует такой сдвиг $0 \leqslant p < n$, что $s_{((i+p-1) \bmod n)+1} = t_i$ для всех $1 \leqslant i \leqslant n$ (другими словами, существует целое число $0 \leqslant p < n$, что если сдвинуть перестановку s вправо p раз, то мы получим перестановку t).

Изначально перестановки s и t отличаются, и ваша задача сделать их равными. Для этого вы можете применять следующую операцию с перестановкой s:

- Выбрать целое число $1 \leqslant b \leqslant n$.
- Совершить b **прыжков** с элементом, равным b. Во время каждого прыжка элемент со значением b перемещается на одну позицию вперёд, меняясь местами с элементом, стоящим непосредственно перед ним. Другими словами, пусть i — такой индекс, что $s_i = b$. Во время одного прыжка, если $i < n$, то вам нужно поменять местами элементы s_i и s_{i+1}, а иначе вам нужно поменять местами элементы s_n и s_1.

Вам нужно найти любую последовательность операций $b_1, b_2, \ldots, b_k$, после выполнения которой перестановка s будет равна перестановке t. Вам не нужно минимизировать количество операций, но их количество не должно превышать 10^5.

Формат входных данных

В первой строке входных данных содержится одно целое число n ($3 \leqslant n \leqslant 100$) — количество элементов в перестановках.

Во второй строке входных данных содержится n различных целых чисел $s_1, s_2, \ldots, s_n$ ($1 \leqslant s_i \leqslant n$) — описание перестановки s. Гарантируется, что $s_1 = 1$.

В третьей строке входных данных содержится n различных целых чисел $t_1, t_2, \ldots, t_n$ ($1 \leqslant t_i \leqslant n$) — описание перестановки t. Гарантируется, что $t_1 = 1$.

Гарантируется, что перестановки s и t различны.

Формат выходных данных

Выведите любую последовательность операций, с помощью которой можно получить из перестановки s перестановку t. Каждую операцию нужно выводить в отдельной строке как одно целое число b. Количество операций не должно превышать 10^5.

Можно показать, что при заданных ограничениях ответ существует.

Пример

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

Замечание

Рассмотрим изменение перестановки s после применения операций в примере из условия:

- Изначально $s = [1, 3, 5, 2, 4]$.
- После применения операции для $b = 2$: $s = [2, 3, 5, 4, 1]$.
- После применения операции для $b = 1$: $s = [1, 3, 5, 4, 2]$.

Задача D. След латинского квадрата

Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В этой задаче мы работаем с латинскими квадратами и следами матриц.

Латинский квадрат — это квадратная матрица размером $n \times n$, в которой каждая клетка содержит одно из n различных значений, причём в каждой строке и каждом столбце значения различны. В данной задаче мы имеем дело только с *естественными латинскими квадратами*, в которых используются целые числа от 1 до n.

След квадратной матрицы — это сумма элементов на главной диагонали. Главная диагональ — это диагональ, идущая из левого верхнего угла матрицы в правый нижний.

Вам даны два целых числа n и k. Ваша задача построить любой *естественный латинский квадрат* размера $n \times n$ со следом, равным k, либо сообщить, что это невозможно.

Для $n = 3$ и $k = 6$, например, существуют следующие два возможных решения (элементы главной диагонали подчёркнуты):

$$\begin{matrix} \underline{2} & 1 & 3 \\ 3 & \underline{2} & 1 \\ 1 & 3 & \underline{2} \end{matrix} \qquad \begin{matrix} \underline{3} & 1 & 2 \\ 1 & \underline{2} & 3 \\ 2 & 3 & \underline{1} \end{matrix}$$

Формат входных данных

В единственной строке входных данных содержится два целых числа n и k ($2 \leqslant n \leqslant 50$, $n \leqslant k \leqslant n^2$).

Формат выходных данных

Выведите «`IMPOSSIBLE`», если решения не существует, и «`POSSIBLE`» в противном случае. Во втором случае дополнительно выведите n строк по n целых чисел в каждой, представляющих естественный латинский квадрат со следом, равным k.

Если существует несколько вариантов ответа, вы можете вывести любой из них.

Примеры

стандартный ввод	стандартный вывод
3 6	POSSIBLE 1 3 2 3 2 1 2 1 3
2 3	IMPOSSIBLE

Задача E. Остаток суммы равен длине

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вам даны массив целых чисел $a_1, a_2, \ldots, a_n$ и натуральное число k.

Ваша задача посчитать количество непустых подотрезков массива $1 \leqslant l \leqslant r \leqslant n$, таких что $(a_l + a_{l+1} + \ldots + a_r) \bmod k = r - l + 1$. Другими словами, остаток от деления суммы чисел на подотрезке на число k должен быть равен длине подотрезка.

Формат входных данных

В первой строке входных данных содержится два целых числа n и k ($1 \leqslant n \leqslant 2 \cdot 10^5$, $1 \leqslant k \leqslant 10^9$) — длина массива a и число k из условия задачи.

Во второй строке входных данных содержится n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^9$).

Формат выходных данных

Выведите одно целое число — количество подходящих подотрезков.

Примеры

стандартный ввод	стандартный вывод
6 4 1 3 5 6 2 4	3
9 4 2 2 4 2 4 2 4 2 4	7

Замечание

В первом тестовом случае подходят следующие три подотрезка:

- $l = 1$ и $r = 1$. В этом случае сумма чисел на подотрезке равна 1, длина подотрезка равна 1 и верно, что $1 \bmod 4 = 1$.
- $l = 3$ и $r = 3$. В этом случае сумма чисел на подотрезке равна 5, длина подотрезка равна 1 и верно, что $5 \bmod 4 = 1$.
- $l = 5$ и $r = 6$. В этом случае сумма чисел на подотрезке равна $2 + 4 = 6$, длина подотрезка равна 2 и верно, что $6 \bmod 4 = 2$.

Задача F. Нанопроцессор

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В рамках проекта по обратному проектированию передовых наночипов группа студентов ФПМИ МФТИ получила задание восстановить архитектуру секретного нанопроцессора, разработанного в одном из ведущих научных центров. К сожалению, сам чип запечатан в термостойкий наноконтейнер, и прямое изучение его структуры невозможно.

Однако с помощью методов неразрушающего тестирования, включая анализ с использованием чёрного наноящика, удалось получить косвенную информацию о внутреннем устройстве процессора.

Чип представляет собой прямоугольную сетку размером $n \times m$ ячеек. В каждой ячейке расположен один из двух типов наноэлементов:

- **ботвизатор** — элемент размером 1×1,
- **расквантизатор** — элемент размером $h \times 1$, всегда ориентированный вертикально, то есть занимающий h ячеек в одном столбце.

Все элементы полностью заполняют сетку — свободных ячеек нет. Известны следующие данные:

- Для каждой строки i ($1 \leqslant i \leqslant n$) известно количество ботвизаторов в этой строке — a_i.
- Для каждого столбца j ($1 \leqslant j \leqslant m$) известно количество расквантизаторов, чья верхняя ячейка находится в этом столбце — b_j. Каждый расквантизатор учитывается ровно один раз — по своему верхнему положению.

Ваша задача — по этим данным восстановить возможную конфигурацию чипа или определить, что предоставленные данные противоречивы и такая конфигурация невозможна.

Формат входных данных

В первой строке входных данных содержится три целых числа n, m, h ($1 \leqslant n, m \leqslant 500$, $1 \leqslant h \leqslant n$) — число строк и столбцов в сетке и высота расквантизатора.

Во второй строке входных данных содержится n целых чисел $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i \leqslant m$), где a_i — количество ботвизаторов в i-й строке сетки.

В третьей строке входных данных содержится m целых чисел $b_1, b_2, \ldots, b_m$ ($0 \leqslant b_j \leqslant n$), где b_j — количество расквантизаторов, начинающихся в j-м столбце.

Формат выходных данных

Если восстановить структуру чипа невозможно, выведите единственное слово «`inconsistent`». В противном случае выведите n строк по m символов — реконструированное изображение чипа, где:

- `*` — ботвизатор,
- `+` — верхняя или нижняя ячейка расквантизатора,
- `|` — внутренняя (не крайняя) ячейка расквантизатора.

Если существует несколько решений, выведите любое из них.

Примеры

стандартный ввод	стандартный вывод
4 3 3 2 1 1 2 1 1 0	+** |+* +|* *+*
3 3 2 1 1 1 1 1 1	inconsistent

Задача G. Четкая лаборатория

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Четкая лаборатория ФПМИ известна своими «`Just Odd Inventions`». Недавно они разработали текстовый редактор под названием «`Just Odd Editor`».

Редактор поддерживает следующие операции. Пусть x — строка, отображаемая на экране редактора, а y — содержимое буфера обмена. Изначально строки x и y пустые.

- **Операция A.** Добавить один символ в конец строки x. В этом случае выбирается произвольный символ c, и x заменяется на $x + c$.
- **Операция B.** Выделить и вырезать всю строку. Тогда y становится равным x, после чего x очищается (строка x становится пустой).
- **Операция C.** Вставить содержимое буфера обмена в конец строки x. Тогда x заменяется на $x + y$.

Выполнение каждой операции требует времени: операция **A** занимает a единиц времени, операция **B** — b единиц времени, операция **C** — c единиц времени.

Вам даны целое число n, строка s длины n, а также целые числа a, b и c. Ваша задача найти наименьшее время, необходимое для того, чтобы строка x стала равна строке s.

Формат входных данных

В первой строке входных данных содержится одно целое число n ($1 \leqslant n \leqslant 2500$) — длина строки s.

Во второй строке входных данных содержится строка s длины n, состоящая из строчных букв латинского алфавита.

В следующих трёх строках входных данных содержатся целые числа a, b и c ($1 \leqslant a, b, c \leqslant 10^9$) — время выполнения соответствующих операций.

Формат выходных данных

Выведите одно целое число — минимальное время, необходимое для того, чтобы строка x стала равна строке s.

Пример

стандартный ввод	стандартный вывод
15 abacabadabacaba 10 5 2	67

Задача H. Panely

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В новой стартап-компании «Панели» отказались от традиционной корпоративной иерархии с менеджерами и подчинёнными. Вместо этого была принята свободная корпоративная культура, где каждый сотрудник, называемый «панелер», сам за себя и может свободно выбирать наставника.

В компании есть n панелеров, которые пронумерованы целыми числами от 1 до n. Изначально каждый панелер выбрал себе одного наставника, с которым начал обсуждать рабочие проекты. Панелер с номером i выбрал себе в качестве наставника панелера с номером a_i. Отметим, что панелеры не могут выбирать самих себя в качестве наставника. Другими словами, $a_i \neq i$. Также в дальнейшем будем называть панелера x подопечным панелера y, если y является наставником x.

У этого свободного выбора наставников очень быстро появились следующие две проблемы:

1. Некоторые панелеры становились слишком популярными и получали слишком много подопечных, из-за чего у них не оставалось времени на собственную работу.
2. Образовывались изолированные группы (например, если два панелера выбирали в качестве наставников друг друга), а такие группы были не связаны с остальной компанией.

Чтобы исправить эти недостатки, было коллективно решено:

1. Каждый панелер должен быть наставником **ровно одного** другого панелера.
2. Если считать, что каждый панелер общается только со своим наставником и со своим подопечным, то информация должна быть способна дойти от любого панелера до любого другого (граф наставничества должен быть связным).

Чтобы поощрить более опытных сотрудников (сотрудников с наименьшими номерами), было решено использовать следующее правило при выборе окончательного распределения наставников:

- Если сравниваются два возможных корректных распределения, то находим панелера с минимальным номером i, для которого назначения различаются.
- Если в одном из распределений панелер i сохраняет своего исходного наставника, то предпочтительнее это распределение.
- Если же панелер i в обоих распределениях получает нового наставника, то предпочтительнее будет то распределение, где номер нового наставника меньше.

Формат входных данных

В первой строке входных данных содержится одно целое число n ($2 \leqslant n \leqslant 5 \cdot 10^5$) — количество панелеров.

Во второй строке входных данных содержится n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant n$, $a_i \neq i$), где a_i — номер текущего наставника панелера i.

Формат выходных данных

Выведите n целых чисел $b_1, b_2, \ldots, b_n$ — новое распределение наставников, удовлетворяющее новым условиям и наиболее предпочтительное среди всех возможных распределений.

Примеры

<table>
<tr><th>стандартный ввод</th><th>стандартный вывод</th></tr>
<tr><td>4
2 1 4 3</td><td>2 3 4 1</td></tr>
<tr><td>3
3 3 1</td><td>3 1 2</td></tr>
</table>

Задача I. Квартира для Дениса

Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Денис нашел свою первую работу и теперь ищет квартиру. Для него важно жить близко к друзьям, но при этом как можно дальше от родителей.

У Дениса есть n друзей, которых мы для удобства пронумеровали целыми числами от 1 до n. Друг с номером i живёт в точке (x_i, y_i). Чтобы жить близко к другу с номером i, квартира Дениса должна находиться на расстоянии не более r_i от места жительства i-го друга. Родители Дениса живут в точке $(0, 0)$.

Ваша задача определить, на каком максимальном расстоянии от родителей может поселиться Денис, оставаясь при этом достаточно близко ко **всем** своим друзьям. Обратите внимание, что Денис может выбрать квартиру в точке с не целочисленными координатами.

Гарантируется, что существует хотя бы одна точка на координатной плоскости, в которой Денис может выбрать квартиру и быть близко ко всем своим друзьям.

Формат входных данных

В первой строке входных данных содержится одно целое число n ($1 \leqslant n \leqslant 50$) — количество друзей Дениса.

В следующих n строках входных данных описывается информация о друзьях Дениса. Каждая строка содержит три целых числа x_i, y_i и r_i ($-10^3 \leqslant x_i, y_i \leqslant 10^3$, $1 \leqslant r_i \leqslant 10^3$), где (x_i, y_i) — координаты местожительства i-го друга, а r_i — максимально допустимое расстояние, на котором Денис готов жить от этого друга.

Формат выходных данных

Выведите одно вещественное число — максимально возможное расстояние от родителей, удовлетворяющее всем условиям.

Ваш ответ будет считаться правильным, если абсолютная или относительная погрешность не превосходит 10^{-9}.

Формально, пусть ваш ответ равен a, а ответ жюри равен b. Ваш ответ будет считаться правильным, если и только если $\frac{|a-b|}{\max(1,|b|)} \leqslant 10^{-9}$.

Примеры

стандартный ввод	стандартный вывод
4 2 0 2 0 2 2 -2 0 2 0 -2 2	0
2 -1 0 100 2 0 100	99.989999500949990

Задача J. Красота расположения книг

Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Демид решил переставить книги на своей очень большой полке, но, чтобы не переутомиться, он планирует поменять местами ровно две книги.

Каждая книга обозначена одной цифрой, i-я книга обозначена цифрой a_i. Демид определяет красоту расположения книг как число, которое получится, если эти цифры записать по порядку. Например, если на полке находятся три книги, которые обозначены цифрами $a_1 = 7$, $a_2 = 5$ и $a_3 = 2$, то красота расположения книг будет равна 752.

Помогите Демиду выбрать две книги для обмена, чтобы после этого красота расположения была максимально возможной. Обратите внимание, что вам нужно обязательно поменять какие-либо две книги местами, даже если это сделает ответ хуже.

Формат входных данных

В первой строке входных данных содержится одно целое число n ($2 \leqslant n \leqslant 10^6$) — количество книг на полке Демида.

Во второй строке входных данных содержится n цифр $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i \leqslant 9$), разделённых пробелами — цифры на книгах, в том порядке, в котором они стоят на полке.

Гарантируется, что первая цифра не равна нулю (другими словами, $a_1 \neq 0$) и как минимум две цифры не равны нулю.

Формат выходных данных

В единственной строке выходных данных выведите n цифр, разделённых пробелами: цифры, написанные на книгах, после того как вы поменяете две книги местами, чтобы получить максимально красивое расположение книг.

Примеры

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

Задача L. Доставка по кампусу

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Студенты Московского Физико-Технического Института активно участвуют в научных проектах и олимпиадах, и часто им нужно обмениваться материалами между лабораториями, расположенными в разных корпусах кампуса. Кампус условно разделён на n зон, пронумерованных целыми числами от 1 до n, в которых расположены аудитории, лаборатории и общежития.

Для перемещения между зонами доступны два типа переходов: наземные переходы и экспресс-туннели. Имеется r наземных переходов, каждый из которых соединяет зоны u_i и v_i и требует w_i энергетических единиц для преодоления. Наземные переходы двусторонние, и их стоимость всегда неотрицательна.

Также доступно p экспресс-туннелей, каждый из которых позволяет переместиться только из зоны a_i в зону b_i за c_i энергетических единиц. Стоимость может быть отрицательной, что соответствует «`энергетическому выигрышу`» — например, эскалатору вниз или автоматической тележке.

Из-за особенностей архитектуры кампуса, если существует экспресс-туннель из зоны a_i в зону b_i, то не существует никакого пути обратно из зоны b_i в зону a_i (ни по наземным переходам, ни по экспресс-туннелям, ни по обоим типам переходов) — это связано с новыми правилами безопасности после эксперимента с вакуумным поездом.

Группа студентов стартует из зоны s и должна доставить пакет с решением задачи в каждую из зон кампуса. Они хотят минимизировать затраты энергии для каждого пункта назначения отдельно.

Ваша задача — определить минимальные энергозатраты на доставку в каждую зону кампуса из стартовой зоны s. Если доставка в какую-либо зону невозможна, следует сообщить об этом.

Формат входных данных

В первой строке входных данных содержится четыре целых числа n, r, p и s ($1 \leqslant s \leqslant n \leqslant 10^5$, $1 \leqslant r, p \leqslant 2 \cdot 10^5$) — количество зон в кампусе, количество наземных переходов, количество экспресс-туннелей и номер зоны, из которой стартует группа студентов.

В следующих r строках входных данных описываются наземные переходы. Каждая строка содержит три целых числа u_i, v_i и w_i ($1 \leqslant u_i \neq v_i \leqslant n$, $0 \leqslant w_i \leqslant 10^4$).

В следующих p строках входных данных описываются экспресс-туннели. Каждая строка содержит три целых числа a_i, b_i и c_i ($1 \leqslant a_i \neq b_i \leqslant n$, $-10^4 \leqslant c_i \leqslant 10^4$).

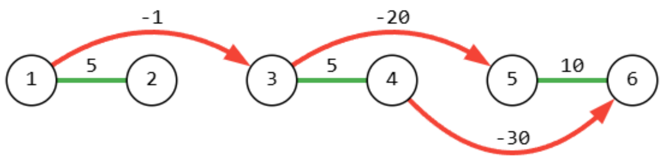
Формат выходных данных

Выведите n строк. В i-й строке выведите минимальную стоимость доставки из зоны s в зону i. Если зона i недостижима из зоны s, выведите «`NO`».

Пример

стандартный ввод	стандартный вывод
6 3 3 4 1 2 5 3 4 5 5 6 10 3 5 -20 4 6 -30 1 3 -1	NO NO 5 0 -20 -30

Замечание



Задача M. Пробежка студентов ФПМИ

Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

На ФПМИ решили устроить спортивный забег по бесконечной прямой аллее. В забеге участвуют n студентов, каждый из которых стартует в своей уникальной позиции x_i и бежит с постоянной скоростью v_i единиц в минуту. Все студенты бегут в сторону увеличения координаты.

Аллея условно разделена на «дорожки» (беговые полосы), чтобы студенты могли обгонять друг друга. Каждый студент в течение всего забега бежит по одной дорожке. Организаторы не хотят, чтобы студенты меняли скорость или оказывались в одной координате во время забега. Другими словами, никакие два студента, бегущие по одной дорожке, не должны одновременно находиться в одной и той же координате в какой-либо момент времени.

Вам нужно определить минимальное количество дорожек, достаточное для того, чтобы все студенты могли бежать в течение t минут без столкновений внутри одной полосы. Обратите внимание, что в момент времени t никакие два студента, бегущие по одной дорожке, также не могут находиться в одной координате.

Формат входных данных

В первой строке входных данных содержится два целых числа n и t ($1 \leqslant n \leqslant 10^5$, $1 \leqslant t \leqslant 10^9$) — количество студентов и длительность забега в минутах.

В следующих n строках входных данных описывается информация о студентах. Каждая строка содержит два целых числа x_i и v_i ($0 \leqslant x_i \leqslant 10^9$, $1 \leqslant v_i \leqslant 10^9$) — стартовая позиция и скорость студента с номером i.

Гарантируется, что все студенты стартуют из различных позиций (другими словами, все x_i различны), а также что студенты даны в порядке возрастания их стартовой позиции (другими словами, $x_1 < x_2 < \ldots < x_n$).

Формат выходных данных

Выведите одно целое число — минимальное количество дорожек, необходимых для организации забега.

Пример

стандартный ввод	стандартный вывод
5 3 0 1 1 2 2 3 3 2 6 1	3

Замечание

В тесте из условия можно распределить студентов по трём дорожкам следующим образом:

- По первой дорожке могут бежать первый и третий студенты.
- По второй дорожке могут бежать второй и четвёртый студенты.
- По третьей дорожке может бежать пятый студент.

Отметим, что в этом примере следующие пары студентов не могут бежать по одной дорожке:

- Третий и четвёртый.
- Четвёртый и пятый.
- Третий и пятый.

Задача N. Цензура

Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

На ФПМИ появился онлайн-редактор текста, в котором студенты пишут статьи и заметки. Администратор заметил, что в тексте встречаются нежелательные слова и решил удалить их автоматически.

Пусть у нас есть исходная строка s. Также есть список запрещённых слов $t_1, t_2, \ldots, t_n$, которые нужно убрать из s.

Процесс удаления следующий:

- Пока в s встречается хотя бы одно слово из списка $t_1, \ldots, t_n$, администратор находит **ближайшее вхождение слова** (с наименьшим индексом в строке s) и удаляет его.
- После удаления слова могут появиться новые вхождения запрещённых слов, которые нужно тоже удалить.

Гарантируется, что никакое слово из списка не является подстрокой другого слова из списка, поэтому каждое вхождение определяется однозначно.

Ваша задача — определить финальное содержание строки s после всех удалений.

Формат входных данных

Первая строка содержит исходную строку s ($1 \leqslant |s| \leqslant 10^5$).

Вторая строка содержит число n ($1 \leqslant n \leqslant 10^5$) — количество слов для удаления.

Следующие n строк содержат слова $t_1, t_2, \ldots, t_n$, каждое из которых состоит только из строчных латинских букв. Суммарная длина всех слов не превышает 10^5.

Формат выходных данных

Выведите строку s после удаления всех запрещённых слов. Гарантируется, что результат не пустой.

Пример

стандартный ввод	стандартный вывод
ffpfpmifpmifmpiq 2 fpmi fpf	fmpiq

Замечание

В тестовом случае выше процесс удаления слов устроен следующим образом:

- Изначально строка равна `ffpfpmifpmifmpiq`.
- После первого удаления строка будет равна `fpmifpmifmpiq`.
- После второго удаления строка будет равна `fpmifmpiq`.
- После третьего удаления строка будет равна `fmpiq`.