

## Задача А. Крестики-нолики

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    0.5 секунд  
Ограничение по памяти:      256 мегабайт

Петя с Васей играют в крестики-нолики. Петя играет крестиками и ходит первым, Вася играет ноликами и ходит вторым. Чтобы сделать игру более интересной, Петя предложил играть на поле размером  $n \times m$ . В остальном правила обычные: игроки ходят по очереди, за ход можно поставить свою фигуру на одну из свободных клеток поля, побеждает тот, кто первым выложит по вертикали, горизонтали или диагонали ряд **из трёх** своих фигур подряд. Помогите Васе понять, сможет ли он на поле данного размера выиграть или хотя бы добиться ничьей при оптимальной игре обоих.

### Формат входных данных

В первой строке записано целое число  $n$  ( $3 \leq n \leq 10^6$ ). Во второй строке записано целое число  $m$  ( $3 \leq m \leq 10^6$ ).

### Формат выходных данных

Выведите «yes», если Вася может выиграть или добиться ничьей, «no» — в противном случае.

### Примеры

стандартный ввод	стандартный вывод
4 4	no
3 3	yes

## Задача В. Олимпиадная задача

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

На олимпиаде по математике Васе встретилась следующая задача: **найти наименьшее число, десятичная запись которого содержит только единицы и которое при этом делится без остатка на 7.**

Вася решил перебрать по очереди числа 1, 11, 111, 1111 и так далее, пока не найдётся число, которое делится на 7.

Число	Остаток от деления на 7
1	1
11	4
111	6
1111	5
11111	2
111111	0

Получается, что ответом задачи является число 111111.

По дороге с олимпиады домой Вася задумался, как бы решалась задача, если бы число записывалось не единицами, а другими одинаковыми цифрами. Более строго, как найти наименьшее положительное целое число, в десятичной записи которого есть только цифры  $k$  и которое делится без остатка на 7. И вообще для любого ли  $k$  такая задача имеет решение?

### Формат входных данных

В единственной строке записано целое число  $k$  ( $1 \leq k \leq 9$ ).

### Формат выходных данных

Если существуют положительные целые числа, в десятичной записи которых есть только цифры  $k$  и которые делятся без остатка на 7, выведите **наименьшее из таких чисел**. Если таких чисел не бывает, выведите  $-1$ .

### Пример

стандартный ввод	стандартный вывод
1	111111

## Задача С. Квадратное уравнение

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Даны рациональные числа  $x_1$  и  $x_2$ , записанные в виде обыкновенных дробей  $\frac{m}{n}$  и  $\frac{p}{q}$  (не обязательно несократимых).

Вам нужно придумать такое квадратное уравнение с целыми коэффициентами  $ax^2 + bx + c = 0$ , чтобы числа  $x_1$  и  $x_2$  являлись его корнями.

### Формат входных данных

Входные данные состоят из четырёх строк с целыми числами  $m$ ,  $n$ ,  $p$  и  $q$  ( $-10^6 \leq m, p \leq 10^6$ ;  $1 \leq n, q \leq 10^6$ ). Числа следуют именно в таком порядке, каждая строка содержит в точности одно число.

### Формат выходных данных

В единственной строке через пробел выведите целые числа  $a$ ,  $b$ ,  $c$  — коэффициенты искомого уравнения. Числа должны удовлетворять ограничениям  $|a|, |b|, |c| \leq 10^{18}$ ;  $a \neq 0$ .

Для любого набора  $m, n, p, q$  эта задача имеет несколько решений, вы можете вывести любое из них.

### Примеры

стандартный ввод	стандартный вывод
1 2 -2 1	2 3 -2
0 20 0 20	1 0 0
1 1 -1 1	1 0 -1

### Замечание

В первом примере уравнение  $2x^2 + 3x - 2 = 0$  имеет корни  $\frac{-3 \pm \sqrt{3^2 + 4 \cdot 2 \cdot 2}}{2 \cdot 2} = \frac{-3 \pm 5}{4}$ , то есть 0.5 и  $-2$ .

Во втором примере уравнение  $x^2 = 0$  имеет два корня 0. Уравнение  $x = 0$  является неверным ответом для этого примера, поскольку не удовлетворяет ограничению  $a \neq 0$ .

В третьем примере уравнение  $x^2 - 1 = 0$  имеет корни 1 и  $-1$ .

## Задача D. Аккорды

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Петя устроился работать в Контур и узнал, что там есть музыкальная комната. Чтобы проводить свои вечера на работе с пользой, Петя решил научиться играть на гитаре. Наш герой — сильный программист, но музыкант он пока ещё начинающий, из-за этого ему бывает неудобно играть аккорды, которые указаны в песне. Помогите Пете написать программу, которая будет сдвигать все аккорды в удобную для него тональность.

Песня представляет собой текст с аккордами, на чётных строках идёт текст, на нечётных — аккорды. Программа должна сместить аккорды на  $X$  полутонов (если  $X$  положительное, то выше, а если  $X$  отрицательное, то ниже). Аккорды циклически смещаются вверх на один полутоном по следующему правилу:

$C \rightarrow C\# \rightarrow D \rightarrow D\# \rightarrow E \rightarrow F \rightarrow F\# \rightarrow G \rightarrow G\# \rightarrow A \rightarrow A\# \rightarrow B \rightarrow C$ .

### Формат входных данных

В первой строке записаны целые числа  $n$  и  $X$  — количество строк, которые следуют далее, и смещение аккордов соответственно ( $2 \leq n \leq 50$ ;  $n$  чётное;  $-12 \leq X \leq 12$ ). Далее в  $n$  строках записан текст песни с аккордами: в нечётных строках записаны аккорды через пробел, в чётных — сам текст. Суммарная длина текста вместе с аккордами не превышает 1 000 символов.

Гарантируется что аккорды могут быть только следующими: C, C#, D, D#, E, F, F#, G, G#, A, A#, B.

### Формат выходных данных

Выведите исходный текст песни вместе со смещёнными аккордами в таком же формате, как во входных данных (количество строк выводить не нужно).

### Примеры

стандартный ввод
8 2 F C I said, ooh, I'm blinded by the lights D# A# No I can't sleep here 'til I feel your touch G D I said, ooh, I'm drowning in the night F# C# Oh when I'm like this, you're the one I trust (hey, hey, hey)
стандартный вывод
G D I said, ooh, I'm blinded by the lights F C No I can't sleep here 'til I feel your touch A E I said, ooh, I'm drowning in the night G# D# Oh when I'm like this, you're the one I trust (hey, hey, hey)

стандартный ввод
4 -3 E can't stop addicted to the shindig C chop Top he says I'm gonna win big
стандартный вывод
C# can't stop addicted to the shindig A chop Top he says I'm gonna win big

## Задача E. Суффиксный массив

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вася не прошёл на летнюю стажировку в Яндекс из-за того, что не решил на собеседовании несложную задачку на строки. Чтобы попасть на стажировку в следующем году, Вася собирается потратить ближайший год на изучение алгоритмов на строках.

Первым делом Вася хочет изучить структуру данных «Суффиксный массив», которая позволяет решать задачи на строках гораздо эффективнее. Алгоритм построения такого массива, который Вася нашёл в книжке, показался ему очень сложным. Единственное, что он понял — это что для каждой пары соседних суффиксов строки нужно найти их наибольший общий префикс. Недолго думая, Вася захотел самостоятельно формализовать и решить эту задачу.

По мнению Васи, строка — это последовательность символов  $(c_1, c_2, \dots, c_n)$ , где  $n$  — длина строки. Суффикс с номером  $i$  ( $1 \leq i \leq n$ ) — это последовательность символов  $(c_i, c_{i+1}, \dots, c_n)$ , являющаяся строкой длины  $n - i + 1$ . Всего есть  $n - 1$  пара соседних суффиксов: первая пара — это 1-й и 2-й суффиксы, вторая пара — это 2-й и 3-й суффиксы, и т. д. Пара суффиксов номер  $i$  — это  $i$ -й и  $(i + 1)$ -й суффиксы.

Наибольший общий префикс пары строк — это наибольшее количество символов в начале этих строк, которые совпадают. Например, наибольший общий префикс строк «pineapple» и «pinescone» имеет длину 4, а наибольший общий префикс строк «pineapple» и «banana» имеет длину 0.

Осталось реализовать алгоритм — для каждой пары соседних суффиксов определить длину наибольшего общего префикса. Васе кажется, что реализация, которую он прочитал в книжке, чересчур длинная и что эту задачу можно решить гораздо проще.

### Формат входных данных

В первой строке дано целое число  $n$  — длина строки ( $2 \leq n \leq 2 \cdot 10^5$ ). Во второй строке дана последовательность из  $n$  строчных букв латинского алфавита, не разделённых пробелами — сама строка.

### Формат выходных данных

Выведите  $n - 1$  строку. В  $i$ -й строке должно быть записано единственное целое число — длина наибольшего общего префикса  $i$ -го и  $(i + 1)$ -го суффиксов.

### Примеры

стандартный ввод	стандартный вывод
9 pineapple	0 0 0 0 0 1 0 0
6 aaabbb	2 1 0 2 1

## Задача F. Серия пенальти

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

В финале чемпионата мира по футболу сборная России встречалась со сборной Франции. Основное и дополнительное время матча закончились со счётом 0:0, а в серии послематчевых пенальти сборная России победила со счётом  $r:f$ .

Приведите пример серии пенальти, при которой мог получиться такой итоговый счёт.

Правила серии послематчевых пенальти:

1. Обе команды выполняют удары поочерёдно. Команда, которая производит удар первой, определяется с помощью жребия.
2. Обе команды выполняют по пять ударов (за исключением двух ситуаций, описанных ниже). Команда, забившая большее количество голов, становится победителем матча.
3. Как только одна из команд забьёт больше голов, чем могла бы забить другая после выполнения пяти ударов, выполнение ударов прекращается. Например, обе команды выполнили по три удара, счёт равен 3:1. Если игрок выигрывающей команды успешно исполнит четвёртый удар, то серия завершится со счётом 4:1. В самом деле, проигрывающая команда смогла бы забить максимум ещё два гола (и в лучшем для неё случае счёт стал бы 4:3), поэтому нет смысла исполнять дальнейшие удары.
4. Если после выполнения пяти пар ударов обе команды забьют равное количество голов (возможно, ни одного), выполнение ударов продолжится до тех пор, пока одна из команд не забьёт на один гол больше при одинаковом количестве выполненных ударов. Например, после пяти пар ударов счёт был 3:3, во время шестой пары ударов обе команды забили гол (счёт стал 4:4), во время седьмой — обе не забили, во время восьмой — одна команда забила, а другая нет. Тогда серия завершается после восьмой пары ударов со счётом 5:4.

### Формат входных данных

В единственной строке записаны целые числа  $r$  и  $f$  ( $0 \leq f < r \leq 100$ ).

### Формат выходных данных

Если не существует серии пенальти с итоговым счётом  $r:f$ , выведите «No solution».

В противном случае выведите любую из возможных серий. В первой строке выведите «Russia» или «France» в зависимости от того, чей игрок нанёс первый удар. Во второй строке выведите последовательность символов «+» и «-» длиной не более 100, описывающую забитые и незабитые игроками **сборной России** голы (в порядке, в котором наносились удары). В третьей строке в том же формате опишите удары игроков **сборной Франции**.

### Примеры

стандартный ввод	стандартный вывод
4 1	Russia ++++ --+
5 4	France +++++ -++++
5 0	No solution
6 4	No solution

## Задача G. Манхэттенские полицейские

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Самое сердце Нью-Йорка, остров Манхэттен, состоит из  $m+1$  авеню и из  $n+1$  улицы, образующих правильную прямоугольную сетку. Авеню — это дорога, проходящая с севера на юг, улица — это дорога, проходящая с запада на восток.

Авеню нумеруются от 0 до  $m$  с запада на восток, улицы нумеруются от 0 до  $n$  с севера на юг. Перекрёсток на пересечении авеню с номером  $a$  и улицы с номером  $b$  обозначается как  $(a; b)$ . Расстояние между любыми соседними авеню и любыми соседними улицами равняется одной у.е. Поскольку на Манхэттене можно передвигаться лишь по дорогам, расстояние от перекрёстка  $(x_1; y_1)$  до перекрёстка  $(x_2; y_2)$  равняется  $|x_1 - x_2| + |y_1 - y_2|$  у.е.

Сегодня Манхэттен оказался буквально заполонён полицейскими — на каждом перекрёстке дежурит ровно один полицейский!

Это беспокоит Дона Рубини, крёстного отца уральской мафии, который скрывается от преследования в своём тайном убежище на перекрёстке  $(x; y)$ . Чтобы сбежать с Манхэттена, ему нужно найти связного. Доподлинно не известно, где находится связной, но известно, что где-то на расстоянии не более  $k$  у.е. от убежища Дона Рубини. Чтобы оценить опасность вылазки в поисках связного, Дон Рубини хочет посчитать количество полицейских, находящихся на расстоянии не более  $k$  у.е. от перекрёстка  $(x; y)$ .

Задача состоит из  $q$  независимых запросов, на каждый из которых нужно отвечать отдельно.

### Формат входных данных

В первой строке записаны целые числа  $m$ ,  $n$  и  $q$  — номер самой восточной авеню, номер самой южной улицы и количество запросов соответственно ( $1 \leq m, n \leq 1\,000$ ;  $1 \leq q \leq 5\,000$ ).

В каждой из следующих  $q$  строк записаны целые числа  $x$ ,  $y$  и  $k$  ( $0 \leq x \leq m$ ;  $0 \leq y \leq n$ ;  $1 \leq k \leq m + n$ ) — координаты убежища Дона Рубини и гарантированное расстояние до связного (в у.е.).

### Формат выходных данных

В каждой из  $q$  строк выведите ответ на очередной запрос — количество полицейских на расстоянии не более чем  $k$  у.е. от убежища Дона Рубини.

### Пример

стандартный ввод	стандартный вывод
5 5 4	23
2 3 3	33
0 4 7	32
5 2 6	4
4 0 1	

## Задача Н. Эхолокация

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Подводная лодка, отправленная на поиски затонувшего города, нашла на дне океана неизвестное строение. Предположительно, это было здание, построенное древней цивилизацией. Подтверждает эту теорию то, что план строения представляет собой  $2n$ -угольник без самопересечений и самокасааний, причём угол между каждой парой смежных сторон составляет ровно  $90$  градусов.

Экипаж подводной лодки ввёл прямоугольную декартову систему координат таким образом, что  $n$  сторон многоугольника в ней расположены горизонтально, а другие  $n$  сторон — вертикально. Более того, в введённой системе координат каждая вершина многоугольника имеет целые координаты.

В арсенале подводной лодки есть эхолокатор, с помощью которого отсканировали все вертикальные стороны. К сожалению, определить положение горизонтальных сторон не удалось.

Экипаж прислал вам отчёт сканирования: было найдено  $n$  вертикальных сторон многоугольника,  $i$ -я из которых имеет концевые вершины в координатах  $(x_i, y_{i1})$  и  $(x_i, y_{i2})$  (естественно, что координаты  $x$  этих двух вершин совпадают). Необходимо определить площадь исходного многоугольника.

### Формат входных данных

В первой строке записано единственное целое число  $n$  — количество вертикальных отрезков ( $2 \leq n \leq 1000$ ).

Далее идут  $n$  строк, каждая из которых содержит через пробел целые числа  $x_i, y_{i1}$  и  $y_{i2}$  — координаты  $i$ -го вертикального отрезка ( $0 \leq x_i \leq 1000$ ;  $0 \leq y_{i1} < y_{i2} \leq 1000$ ). Гарантируется, что никакие два отрезка не имеют общих точек и что все отрезки составляют все вертикальные стороны некоторого корректного  $2n$ -угольника.

### Формат выходных данных

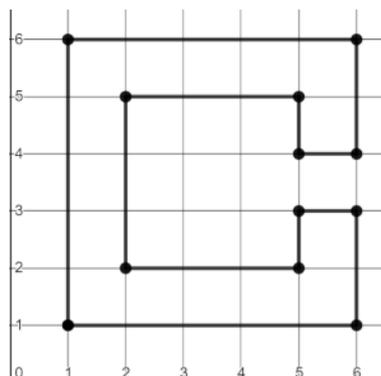
Выведите единственное целое число — площадь исходного многоугольника.

### Пример

стандартный ввод	стандартный вывод
6 1 1 6 6 1 3 6 4 6 2 2 5 5 2 3 5 4 5	15

### Замечание

Иллюстрация к примеру:



## Задача I. Подарок для Вовы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Совсем недавно начался учебный год, а маленький Вова вместе со своими друзьями уже записался на новый кружок олимпиадной математики! Недавно на занятиях проходили мультимножества и строки, поэтому в качестве развлечения на переменах ученики выбирают некоторую строку из строчных латинских букв и вычисляют для неё **мультимножество**  $M$ , состоящее из всех пар соседних символов строки. Так например,  $M(abcbab) = \{(a, b), (b, c), (c, b), (b, a), (a, b)\}$ . Формально, мультимножество  $M(s_1s_2 \dots s_n)$  состоит из пар  $(s_i, s_{i+1})$  для всех  $1 \leq i \leq n-1$ . Каждая пара входит в мультимножество столько раз, сколько раз эта пара символов встречается в строке на соседних позициях именно в таком порядке.

Скоро у Вовы день рождения, и родители захотели подарить ему строку  $s$ . Они уже нашли её в интернет-магазине, но не могут решиться на покупку, ведь Вова будет очень огорчён, если у кого-то из его друзей уже есть строка  $t$  такая, что:

1. мультимножества  $M(t)$  и  $M(s)$  совпадают;
2.  $t$  лексикографически меньше  $s$ .

Чтобы сделать сыну самый лучший подарок, родители хотят по строке  $s$  найти лексикографически наименьшую строку  $t$  такую, что  $M(t) = M(s)$ . Помогите им!

Напомним, что строка  $t$  считается лексикографически меньше строки  $s$ , если существует такое  $\ell$  ( $1 \leq \ell \leq \min(|t|, |s|)$ ), что  $t[1..\ell-1] = s[1..\ell-1]$  и  $t[\ell] < s[\ell]$ , или  $|t| < |s|$  и  $t = s[1..|t|]$ .

### Формат входных данных

В первой строке записано единственное целое число  $n$  — длина строки  $s$  ( $2 \leq n \leq 10^5$ ).

Во второй строке записана строка  $s$  длины  $n$ , состоящая из строчных латинских букв (a-z).

### Формат выходных данных

Выведите лексикографически наименьшую строку  $t$ , для которой мультимножество  $M(t)$  совпадает с мультимножеством  $M(s)$ .

### Примеры

стандартный ввод	стандартный вывод
3 bab	aba
5 хухау	хауху

## Задача J. Приличные слова

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Аня работает в NAUMEN, там она занимается разработкой робота, который будет автоматически генерировать различные тексты.

Сегодня Анин робот сгенерировал слово  $S$  из строчных латинских букв, которое показалось Ане не очень приличным. Благо, у Ани есть база из  $q$  приличных слов (также из строчных латинских букв), причём про каждое слово известно, что любая буква встречается в нём не больше одного раза. Ещё для каждого слова в базе хранится целое число — его неприличность. В базе могут встречаться одинаковые слова, имеющие разную неприличность. Аня может переставлять буквы в любом слове из базы произвольным образом (но не может добавлять, убирать или заменять буквы), при этом неприличность слова не изменяется.

Теперь Аня хочет разбить слово  $S$  на несколько слов из базы (возможно, изменив порядок букв в них) так, чтобы суммарная неприличность этих слов была минимальной. При этом нельзя смешивать буквы разных слов: сначала должны идти все буквы одного слова, затем — все буквы другого слова, и так далее. Одно и то же слово из базы можно использовать сколько угодно раз, при этом в разных использованиях можно переставлять буквы этого слова по-разному.

### Формат входных данных

В первой строке записано непустое слово  $S$  из строчных латинских букв, длиной не более  $10^5$ . Во второй строке записано целое число  $q$  — количество приличных слов в базе Ани ( $1 \leq q \leq 10^5$ ). Далее следуют  $q$  строк,  $i$ -я из которых содержит через пробел строку  $w_i$  и целое число  $c_i$  —  $i$ -е слово из базы и его неприличность ( $1 \leq c_i \leq 10^9$ ). Все слова в базе непустые, состоят только из строчных латинских букв, при этом в каждом слове буквы не повторяются.

### Формат выходных данных

Если невозможно составить слово  $S$ , используя слова из базы Ани и перестановки букв в них, выведите  $-1$ , иначе выведите минимальную суммарную неприличность использованных слов.

### Примеры

стандартный ввод	стандартный вывод
abba 2 ab 5 ba 125	10
ab 3 a 12 c 1 xyz 8	-1

### Замечание

В первом примере можно взять слово «ab» с неприличностью 5, потом ещё раз это же слово, но поменяв в нём буквы местами. Суммарная неприличность составит  $5 + 5 = 10$ .

Во втором примере невозможно получить слово  $S$ , потому что ни одно слово из базы не содержит букву «b».

## Задача К. Тараканы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	256 мегабайт

В комнате, в которой живёт Вася, есть тараканы. А может, это просто грязные пятна на стене? Издалека на стене видно несколько чёрных точек. Каждая из них может быть либо тараканом, либо пятном. Отличаются они тем, что пятно всегда остаётся на месте, а таракан может перемещаться.

Вася ввёл прямоугольную декартову систему координат, в которой его стена занимает квадрат с противоположными вершинами  $(0, 0)$  и  $(10^9, 10^9)$ . Далее он последовательно сделал  $n$  фотографий стены, на  $i$ -й фотографии запечатлев подпрямоугольник стены с противоположными вершинами  $(x_{iL}, y_{iL})$  и  $(x_{iR}, y_{iR})$ . Оказалось, что на  $i$ -й фотографии есть  $k_i$  чёрных точек,  $j$ -я из которых имеет координаты  $(x_{ij}, y_{ij})$  (координаты абсолютные, а не относительно угла фотографии).

Вася считает, что если чёрная точка присутствовала на всех фотографиях, покрывающих её координаты, то это грязное пятно. А если чёрная точка была на одной фотографии, но отсутствовала на другой, на которой должна была быть, то это таракан. Вася также считает, что если таракан появляется несколько раз в одной и той же точке, то это могут быть разные тараканы. Но в одной точке не могут находиться одновременно два таракана или таракан и пятно.

Определите наибольшее возможное количество тараканов на фотографиях Васи.

### Формат входных данных

В первой строке записано единственное целое число  $n$  — количество фотографий ( $1 \leq n \leq 20\,000$ ).

Далее следуют  $n$  групп строк,  $i$ -я группа начинается со строки, содержащей через пробел целые числа  $x_{iL}$ ,  $y_{iL}$ ,  $x_{iR}$ ,  $y_{iR}$  и  $k_i$  — координаты противоположных вершин фотографии и количество чёрных точек на ней ( $0 \leq x_{iL} < x_{iR} \leq 10^9$ ;  $0 \leq y_{iL} < y_{iR} \leq 10^9$ ;  $0 \leq k_i \leq 50\,000$ ). После этой строки идут ещё  $k_i$  строк,  $j$ -я из них содержит через пробел целые числа  $x_{ij}$  и  $y_{ij}$  — координаты очередной чёрной точки ( $x_{iL} \leq x_{ij} \leq x_{iR}$ ;  $y_{iL} \leq y_{ij} \leq y_{iR}$ ). Все чёрные точки на одной фотографии различны.

Гарантируется, что  $k_1 + k_2 + \dots + k_n \leq 50\,000$ .

### Формат выходных данных

Выведите наибольшее возможное количество тараканов на фотографиях.

### Примеры

стандартный ввод	стандартный вывод
3 0 0 10 2 2 1 1 9 1 2 0 8 2 0 0 0 10 2 2 1 1 10 1	2
3 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 0	2

### Замечание

В первом примере в точках  $(9, 1)$  и  $(10, 1)$  находятся тараканы, а в точке  $(1, 1)$  — пятно.

Во втором примере на фотографиях дважды присутствует таракан в точке  $(0, 0)$ , и Вася считает, что это могут быть два разных таракана.