Problem A. Составляем прямоугольник

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 секунда
Memory limit:	256 мегабайт

У Васи есть набор из 4 палочек, они имеют длины a, b, c, d. Вася хочет составить из них прямоугольник, однако он обнаружил, что это возможно сделать не для всех четверок a, b, c, d. Тогда он решил разрезать некоторые палочки на две части, одну из которых он выбросит, а другую использует как сторону прямоугольника. Помогите Васе сделать разрезы так, чтобы у него получился прямоугольник максимальной площади.

Input

В единственной строке заданы 4 натуральных числа a, b, c, d ($1 \le a, b, c, d \le 99$), разделенные пробелами — исходные длины палочек, имеющихся у Васи.

Output

В единственной строке выведите натуральное число s — максимальную возможную площадь прямоугольника, составленного Васей.

Examples

standard input	standard output
2 7 3 8	14
2 2 3 3	6

Note

В первом примере Вася может обрезать третью и четвертую палочки так, чтобы их длины стали 2 и 7 соответственно. Тогда он составит прямоугольник со сторонами 2 и 7, его площадь равна 14.

Problem B. Скобки

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 секунда
Memory limit:	256 мегабайт

Мальчик Вася сегодня нашёл для себя новую игру на просторах Интернета.

Дана строка, состоящая из маленьких английских букв и символов '(' и ')'. Игра состоит из нескольких ходов. На каждом ходу Вася может выбрать подстроку, начинающуюся с открывающей скобки и заканчивающуюся закрывающей, и удалить её из строки. После этого оставшиеся части строки склеиваются, и игра продолжается, пока у Васи есть возможные ходы. Если в конце игры в строке есть хотя бы одна скобка, Вася терпит поражение. Иначе он получается количество очков, равное длине оставшейся строки. Естественно, он не хочет проиграть и желает максимизировать итоговое количество очков.

Например, пусть была задана строка $s = win(ter)comp(u(t)er)school$.

1. Возьмем подстроку (ter). После её удаления останется $wincomp(u(t)er)school$.
2. Возьмем подстроку (t). После её удаления останется $wincomp(uer)school$.
3. Возьмем подстроку (uer). После её удаления останется $wincompschool$.

На первом шаге можно было удалить подстроку $(ter)comp(u(t)$, но тогда осталась бы закрывающая скобка, которую удалить бы уже не получилось. Если сразу удалить подстроку $(ter)comp(u(t)er)$, но останется строка $winschool$ — её длина меньше, чем у $wincompschool$.

Покажите Васе, как нужно делать ходы, чтобы длина итоговой строки была максимально возможной.

Input

В единственной строке файла содержится непустая строка s длиной не более 10^5 символов, состоящая из маленьких букв английского алфавита и символов '(' и ')'. Гарантируется, что существует последовательность ходов, после которой остается строка, не содержащая скобок.

Output

Выведите в единственной строке файла исходную строку, в которой символы, которые будут удалены, заменены на символы '*'.

Examples

standard input	standard output
(i)nt)	******
zanknvn()(t()l	zanknvn******l

Problem C. Поиск пути

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 секунда
Memory limit:	256 мегабайт

На клетчатой бумаге задан квадрат, его левая нижняя вершина имеет координаты $(1,\ 1)$, а правая верхняя — $(n,\ n)$. Соответственно, ось Ox направлена слева направо, а ось Oy — снизу вверх.

Также для каждого $i = 1, 2, \ldots, n$ выбраны две точки с координатами (L_i, i) и (R_i, i) $(1 \le L_i \le R_i \le n)$. Требуется найти длину кратчайшего пути **по линиям сетки**, удовлетворяющего следующим условиям:

- начинается в точке $(1, 1)$;
- заканчивается в точке (n, n);
- не содержит движений вниз;
- проходит через все точки (L_i, i) и (R_i, i).

Путь может проходить по одному и тому же ребру дважды.

Input

В первой строке входного файла задано целое число n — количество строк и столбцов в квадрате $(1 \le n \le 10^6)$. В i-й из последующих n строк заданы два целых числа L_i и R_i $(1 \le L_i \le R_i \le n)$.

Output

В единственной строке выведите целое число — длину кратчайшего пути, удовлетворяющего условию.

Examples

standard input	standard output
7 2 7 3 5 1 4 1 3 3 7 4 6 3 5	34

Problem D. Яблоко от яблони

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 секунда
Memory limit:	256 мегабайт

В Берляндии сорта яблонь нумеруют натуральными числами. Количество яблок, которые даёт яблоня сорта k, определяется так. Рассмотрим группы одинаковых цифр, образующие непрерывный отрезок десятичной записи числа k. Если длина очередной группы равна l, и образована она цифрами d, то добавим к итоговой сумме $d \cdot l^2$.

Например, яблоня сорта 22231170077 даёт $2 \cdot 3^2 + 3 \cdot 1^2 + 1 \cdot 2^2 + 7 \cdot 1^2 + 0 \cdot 2^2 + 7 \cdot 2^2 = 60$ яблок.

От вас требуется посчитать сумму количеств яблок, которые дают яблони сортов $A, A + 1, \ldots, B$.

Input

В первой строке задано натуральное число T ($1 \le T \le 1000$) — количество наборов тестовых данных.

В i-й из следующих T строк заданы два натуральных числа A_i и B_i ($1 \le A_i \le B_i \le 10^{15}$), разделенные пробелом.

Output

Выведите T строк, в i-й из них должно содержаться суммарное количество яблок, которые дают яблони сортов $A_i, A_i + 1, \ldots, B_i$.

Examples

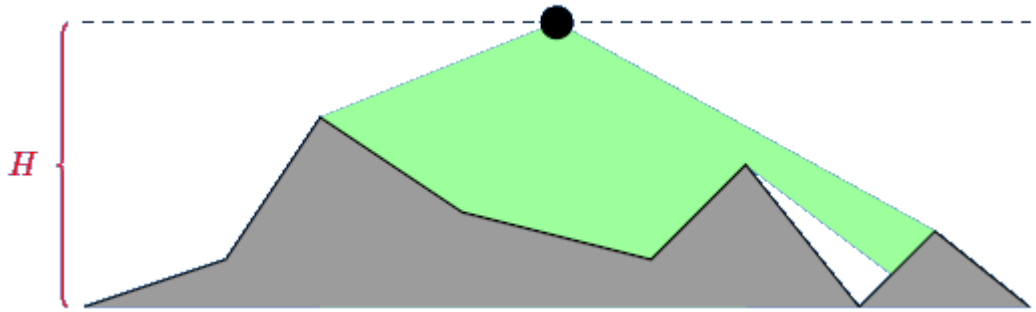
standard input	standard output
2 1 7 14 37	28 172

Problem E. Организация наблюдения

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 секунда
Memory limit:	256 мегабайт

Великая Берляндская Стена состоит из последовательности точек, соединенных отрезками. В некоторых точках построены башни. Над стеной расположена трасса, на которой премьер-министр хочет поставить несколько наблюдательных пунктов так, чтобы из них были видны все башни. Башня видна с наблюдательного пункта, если отрезок, соединяющий наблюдателя и башню, не пересекает ни один из отрезков Стены (при этом он может касаться отрезка Стены или лежать с ним на одной прямой).

Ваша задача — поставить минимальное количество наблюдательных пунктов на трассе так, чтобы каждая башня была видна хотя бы с одного из пунктов.



Input

Первая строка входного файла содержит два натуральных числа n и H ($1 \le n \le 10^5$, $1 \le H \le 10^6$). n — это количество точек, отрезки между которыми образуют Стену. Трасса — это прямая $y = H$.

Следующие n строк описывают Великую Берляндскую Стену. Каждая из этих строк содержит три натуральных числа x_i, y_i, z_i ($0 \le x_i \le 10^6$, $0 \le y_i < H$, $z_i \in \{0, 1\}$). (x_i, y_i) — координаты точки, а $z_i = 1$ тогда и только тогда, когда в этой точке есть башня. Гарантируется, что y_1 и y_n равны 0. Точки даны в порядке возрастания x_i.

Output

В единственной строке выведите натуральное число k — минимальное число наблюдательных пунктов, которые нужно построить на трассе.

Examples

standard input	standard output
9 30 0 0 1 15 5 1 25 20 1 40 10 1 60 5 1 70 15 1 82 0 1 90 8 1 100 0 1	2

Problem F. Множество игр

Input file:	*standard input*
Output file:	*standard output*
Time limit:	0.7 секунд
Memory limit:	256 мегабайт

Игра "Шары и отверстия" играется на прямой. Игра задаётся двумя наборами натуральных чисел $(\{a_1, a_2, \ldots, a_m\}, \{b_1, b_2, \ldots, b_l\})$.

Изначально m шаров ставится в точки с координатами $a_1, a_2, \ldots, a_m$, по одному на каждую точку.

После этого делается l отверстий в точках с координатами $b_1 + 0.5, b_2 + 0.5, \ldots, b_l + 0.5$.

Далее шары толкаются в направлении возрастания координаты, после чего некоторые из них падают в отверстия.

Игрок объявляется победившим, если количество отверстий, в которые попал хотя бы один шар, нечётно.

Вам заданы n множеств $S_1, S_2, \ldots, S_n$. Нужно определить, в каком количестве игр, заданных как (S_i, S_j) для $i < j$, игрок побеждает.

Input

Первая строка содержит натуральное число n $(2 \le n \le 5000)$.

Каждая из следующих n строк содержит натуральное число k_i — размер множества S_i. Далее заданы k_i различных натуральных чисел $S_{i,1}, S_{i,2}, \ldots, S_{i,k_i}$, которые задают множество S_i $(1 \le k_i \le 50, 1 \le S_{i,j} \le 50)$.

Output

Выведите единственное целое неотрицательное число — количество игр, заданных как (S_i, S_j) для $i < j$, в которых игрок побеждает.

Examples

standard input	standard output
2 1 1 2 1 2	1
2 2 1 2 2 2 1	0