Задача A. Футбол

Имя входного файла:	`football.in`
Имя выходного файла:	`football.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В одном большом городе очень любят футбол. На стадионе единственной команды уже давно используются высокие технологии, так, например, в футбольном мяче находится маяк, по местонахождению которого определяется взятие ворот.

Недавно на стадионе было принято решение автоматизировать подсчет времени владения мячом для одного игрока. Для этого и на игроке установили маяк. В какие-то моменты времени оба маяка сообщают о своем положении. Считается, что и игрок, и мяч между сигналами маяков двигаются равномерно и прямолинейно. При этом принято считать, что игрок владеет мячом, если расстояние между ними не более чем R.

Напишите программу, которая по сигналам маяков узнавала бы, сколько времени игрок владел мячом.

Формат входного файла

В первой строке даны два целых числа — N и R ($2 \le N \le 10^5$, $0 \le R \le 10^4$), где N — количество сигналов, полученных от маяков.

Далее следуют N строк, содержащих по пять целых чисел t, x_b, y_b, x_p, y_p каждая, — время получения сигналов и координаты мяча и игрока соответственно ($0 \le t \le 10^9$, $-10^4 \le x_b, y_b, x_p, y_p \le 10^4$). Времена получения сигналов во входном файле возрастают.

Формат выходного файла

В выходной файл выведите одно число — общее время владения мячом. Ответ выводите с максимальной возможной точностью. Проверяющая программа будет сравнивать ответ участника с ответом жюри с точностью до 6 знаков после точки.

Примеры

football.in	football.out
2 1 1 0 0 2 0 2 2 2 0 2	0.5
2 3 10 1 1 3 1 20 3 1 3 3	10
3 2 1 0 0 2 2 2 2 2 0 0 3 2 0 0 2	0.707106781187

Задача B. Квадратный корень из перестановки

Имя входного файла:	`sqroot.in`
Имя выходного файла:	`sqroot.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Перестановкой называется упорядоченный набор N чисел $1, 2, \ldots, N$, где N — порядок перестановки. Каждое число из этого диапазона встречается в перестановке ровно один раз. Перестановку можно рассматривать в виде функции $P\colon \{1 \ldots N\} \to \{1 \ldots N\}$, которая каждому натуральному числу, не превосходящему N, сопоставляет также натуральное число, не превосходящее N, причем различным аргументам сопоставляются различные значения.

Перестановки принято записывать в виде матрицы с двумя строками и N столбцами, первая строка этой матрицы всегда содержит числа от 1 до N в порядке возрастания, а вторая строка содержит числа, сопоставляемые им:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 1 & 5 & 4 & 2 & 6 \end{pmatrix}$$

На перестановках можно определить операцию произведения двух перестановок равного порядка N. Пользуясь способом задания перестановок как функций натуральных чисел, определим произведение так: $(P \cdot Q)(x) = Q(P(x))$, где $1 \le x \le N$. Например,

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 3 & 5 & 6 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 1 & 5 & 4 & 2 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 2 & 6 & 1 \end{pmatrix}$$

Подобно квадратному корню из комплексного числа, можно определить квадратный корень из перестановки. Эта функция будет неоднозначной, поэтому определим квадратный корень из перестановки таким образом:

$$\sqrt{P} = \{Q \mid Q \cdot Q = P\}$$

Ваша задача состоит в том, чтобы по данной перестановке P найти мощность множества $\sqrt{P}$, то есть количество таких перестановок Q, что $Q \cdot Q = P$.

Формат входного файла

В первой строке входного файла содержится число N ($1 \le N \le 100$) — порядок перестановки P.

Во второй строке содержится N целых чисел, описывающих перестановку P. i-тое число в этой строке равно $P(i)$.

Формат выходного файла

Выведите ответ на задачу.

Примеры

sqroot.in	sqroot.out
2 1 2	2
3 3 2 1	0

Задача C. Криминальное чтиво

Имя входного файла:	`pulp.in`
Имя выходного файла:	`pulp.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Иван Днепров пишет детективы. Да, его детективы не отличаются чрезвычайно запутанным сюжетом или особо глубокой и тонкой иронией. Да, один книжный магазин недавно начал продавать его детективы на вес, поскольку логично, что польза от такой книги (количество времени, которое можно убить) прямо пропорциональна количеству страниц в ней и следовательно весу. Но читатели любят его книги, они легко читаются и помогают многим расслабится после трудового дня.

Главный секрет Ивана в том, что на самом деле он не сам изобретает сюжеты для своих детективов. Его брат Алексей придумывает для него идеи, а Иван в свою очередь пишет книги. Разумеется, Иван делится своим заработком с Алексеем. К сожалению, возможности Алексея по придумыванию сюжетов для детективов зависит от его биоритмов. Братья рассчитали таблицу биоритмов Алексея и выяснили, что он в ближайшее время сможет придумать n новых сюжетов, i-й сюжет будет готов на r_i-й день. В один и тот же день Алексей в принципе может придумать несколько сюжетов.

Для каждого сюжета, который Алексей придумает, Иван оценил время, которое ему потребуется чтобы написать детектив. А именно, превращение i-го сюжета в книгу займет p_i дней. У Ивана очень хорошая память, поэтому он может временно отложить написание книги и переключится на другие, а потом вернуться к отложенной.

Конечно, братья хотят чтобы книги были написаны как можно скорее. А именно, они хотят минимизировать среднее время от текущего момента до момента, когда детектив будет готов. Поскольку количество книг фиксировано, это то же самое, что и минимизировать сумму. Если i-я книга будет написана на c_i-й день, то братья хотят минимизировать $\sum c_i$.

По заданным n, r_i и p_i, помогите братьям разработать оптимальное расписание написания детективов.

Формат входного файла

Первая строка входного файла содержит n — количество детективов, которое планируется написать ($1 \le n \le 100\,000$). Следующие n строк содержат по два целых числа: r_i и p_i ($1 \le r_i, p_i \le 10^9$).

Формат выходного файла

Выведите одно число — минимальное возможное значение $\sum c_i$, которого можно достичь.

Пример

pulp.in	pulp.out
2 1 5 2 1	10

У Алексея появляется первая идея в день 1. Иван начинает писать книгу. На второй день у Алексея появляется еще одна идея. Иван переключается на новую книгу, и заканчивает ее через 1 день (на день 3, так что $c_2 = 3$). Теперь Иван возвращается к первой книге и заканчивает ее через 4 дня (1 день он уже над ней работал), это день 7, так что $c_1 = 7$ и $\sum c_i = 10$.

Задача D. Обобщенные скобочные последовательности

Имя входного файла:	`brackets.in`
Имя выходного файла:	`brackets.out`
Ограничение по времени:	3 секунды
Ограничение по памяти:	64 мегабайта

Правильной скобочной последовательностью называется последовательность из $2n$ символов, каждый из которых представляет собой либо «`(`» (открывающую скобку), либо «`)`» (закрывающую скобку), причем выполнены два условия: любой префикс последовательности содержит не меньше открывающих скобок чем закрывающих и во всей последовательности поровну открывающих и закрывающих скобок. Например, последовательности «», «`((()))`», «`()()()()`», «`(())()`» являются правильными скобочными последовательностями, а последовательность «`())(()`» — нет, так как ее префикс «`())`» содержит больше закрывающих скобок чем открывающих, также не является правильной последовательность «`((`», поскольку она содержит разное количество открывающих (две) и закрывающих (ноль) скобок.

Для каждой открывающей скобки можно найти соответствующую ей закрывающую — единственную закрывающую скобку, следующую за ней, такую, что между ними находится правильная скобочная последовательность.

Обобщением правильных скобочных последовательностей являются правильные скобочные последовательности с несколькими типами скобок. Будем рассматривать скобочные последовательности с k типами скобок. Пусть каждая скобка имеет свой *тип* — целое число от 1 до k. Последовательность таких типизированных скобок называется правильной если она во-первых является правильной скобочной последовательностью и во-вторых для каждой открывающей скобки соответствующая ей закрывающая имеет такой же тип. Например, последовательность «$(_1(_2)_2)_1(_1)_1$» является правильной скобочной последовательностью, а последовательность «$(_1(_2)_1)_2(_1)_1$» — нет.

Упорядочим $2k$ различных типизированных скобок некоторым образом. Тогда можно ввести *лексикографический* порядок на скобочных последовательностях длины $2n$ — из двух последовательностей меньше та, в которой при просмотре с начала первый различающийся символ идет раньше во введенном порядке.

Вам заданы n, k, порядок на скобках и правильная скобочная последовательность. Найдите следующую правильную скобочную последовательность в лексикографическом порядке.

Формат входного файла

В этой задаче входной файл содержит несколько тестов. Первая строка входного файла содержит t — количество тестов ($1 \le t \le 10000$). Далее следует описание t тестов.

Первая строка каждого описания содержит два целых числа: n и k ($1 \le n \le 100\,000$, $1 \le k \le 10\,000$). Вторая строка содержит $2k$ целых чисел — перестановку множества $\{-k, -(k-1), \ldots, -2, -1, 1, 2, \ldots, k\}$. Она задает порядок на типизированных скобок. Положительное число i означает открывающую скобку i-го типа, а отрицательное число $-i$ означает закрывающую скобку i-го типа. Третья строка содержит $2n$ целых чисел от $-k$ до k (кроме 0) и задает правильную скобочную последовательность.

Сумма всех n и сумма всех k в каждом входном файле не превышает $100\,000$.

Формат выходного файла

Для каждого теста во входном файле выведите $2n$ целых чисел — описание следующей после заданной в тесте правильной скобочной последовательности. Если последовательность во входном файле последняя в лексикографическом порядке, выведите первую в лексикографическом порядке последовательность.

Пример

brackets.in	brackets.out
2 3 2 1 -1 2 -2 1 2 -2 -1 1 -1 3 2 1 -1 2 -2 1 2 -2 -1 2 -2	1 2 -2 -1 2 -2 1 2 -2 2 -2 -1