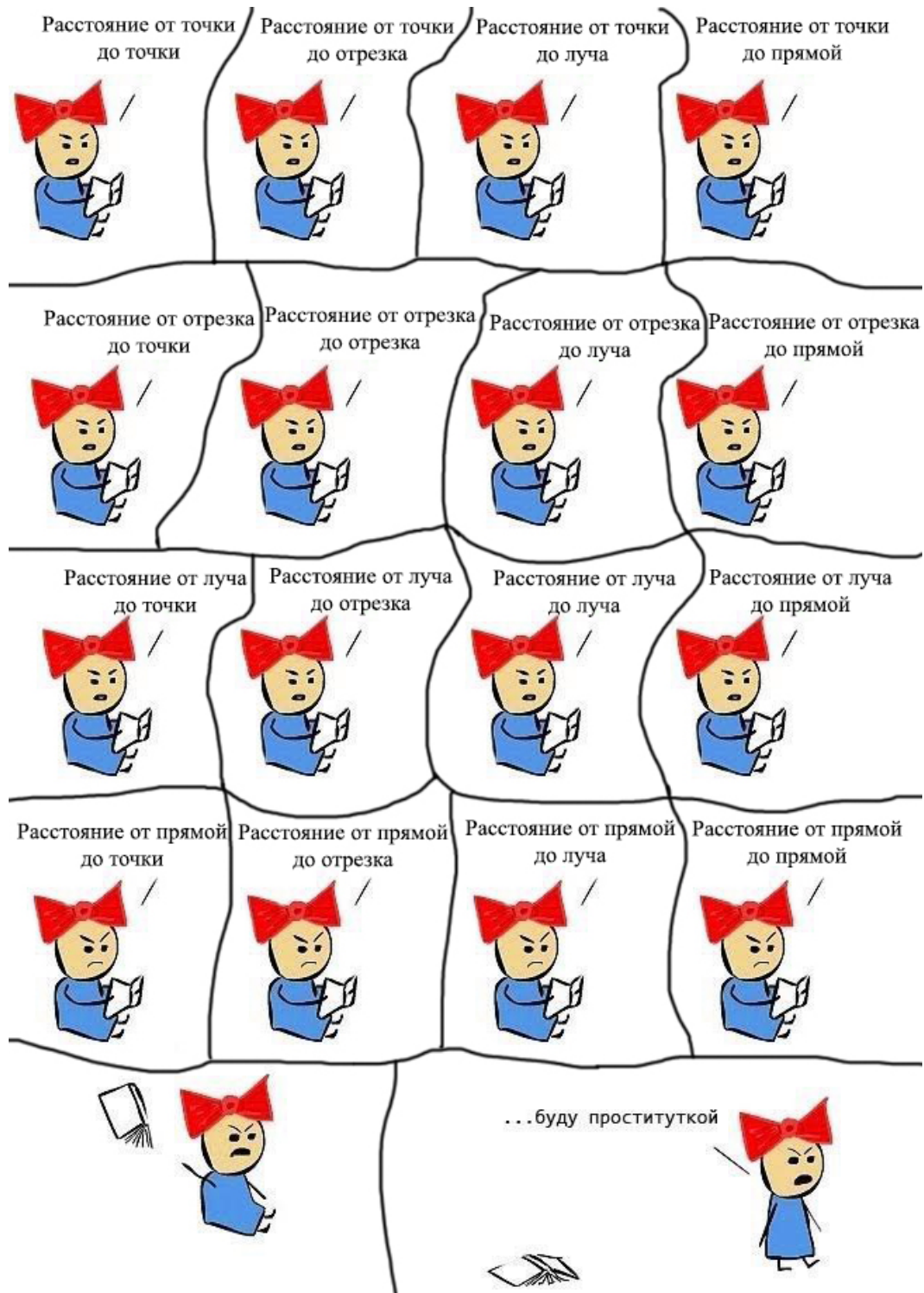


Задача А. 16

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



Даны 4 точки A, B, C, D.

Посчитайте:

- Расстояние от точки A до точки C.
- Расстояние от точки A до отрезка CD.
- Расстояние от точки A до луча CD.
- Расстояние от точки A до прямой CD.
- Расстояние от отрезка AB до точки C.
- Расстояние от отрезка AB до отрезка CD.
- Расстояние от отрезка AB до луча CD.
- Расстояние от отрезка AB до прямой CD.
- Расстояние от луча AB до точки C.
- Расстояние от луча AB до отрезка CD.
- Расстояние от луча AB до луча CD.
- Расстояние от луча AB до прямой CD.
- Расстояние от прямой AB до точки C.
- Расстояние от прямой AB до отрезка CD.
- Расстояние от прямой AB до луча CD.
- Расстояние от прямой AB до прямой CD.

Формат входных данных

Даны координаты четырех точек, по одной точке в строке: $X_a, Y_a, X_b, Y_b, X_c, Y_c, X_d, Y_d$. Все числа целые, по модулю не превосходят 10000.

Формат выходных данных

Выведите 16 чисел по одному в строке. Числа в ответе должны быть выданы с точностью не менее 6 знаков после десятичной точки.

Пример

стандартный ввод	стандартный вывод
1 2	5.6568542495
7 1	5.6000000000
5 6	5.6000000000
8 2	5.6000000000
	4.6031716446
	1.4142135624
	1.4000000000
	1.4000000000
	4.6031716446
	1.1507929111
	0.0000000000
	0.0000000000
	4.6031716446
	1.1507929111
	0.0000000000
	0.0000000000

Задача В. Расстояние между отрезками

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Найдите расстояние между двумя отрезками.

Формат входных данных

В двух строках входного файла даны по четыре целых числа — координаты концов сначала первого, затем второго отрезков. Все числа по модулю не превосходят 10 000.

Формат выходных данных

Одно число — расстояние между отрезками с точностью не менее 10^{-6} .

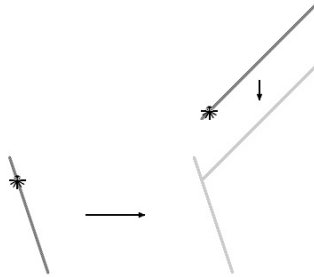
Пример

стандартный ввод	стандартный вывод
1 1 2 2 2 1 3 0	0.70710678

Задача С. Про любовь...

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

Паук и паучиха плывут по озеру на двух веточках. Плавать они не умеют, поэтому смогут встретиться только тогда, когда веточки соприкоснутся.



Считая, что веточки имеют форму отрезков, и что они плывут с постоянными скоростями, определите, сколько осталось ждать встречи несчастным членистоногим.

Формат входных данных

Входной файл содержит 12 чисел: $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, v_{1x}, v_{1y}, v_{2x}, v_{2y}$. Координаты вершин первого отрезка: (x_1, y_1) и (x_2, y_2) , координаты вершин второго отрезка: (x_3, y_3) и (x_4, y_4) , скорость первого отрезка (v_{1x}, v_{1y}) , скорость второго отрезка (v_{2x}, v_{2y}) . Все числа целые и не превосходят по модулю 10^4 . В начальный момент времени веточки не соприкасаются.

Гарантируется, что веточки имеют ненулевую длину.

Формат выходных данных

Выведите в выходной файл время до ближайшего момента, когда веточки соприкоснутся, с ошибкой не более 10^{-4} . Если веточки не соприкоснутся никогда, выведите число -1 .

Примеры

стандартный ввод	стандартный вывод
0 0 -1 3 4 4 7 7 3 0 0 -1	1.6000000000
0 0 -1 3 4 4 7 7 1 0 0 -3	-1

Задача D. Кто ходит в гости по утрам?

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

Пятачок и Винни-Пух каждое утро ходят пить чай в гости к Кролику. Естественно, самым коротким путем.

К сожалению, однажды Винни-Пуху пришла в голову идея вырыть ловушку для Слонопотама. Самое обидное, что они с Пятачком ее даже вырыли. Поэтому теперь каждое утро, идя в гости к Кролику, они боятся в нее провалиться.

Напишите программу, которая посчитает длину самого короткого безопасного пути от домика Винни-Пуха до домика Кролика.

Ловушка для Слонопотама представляет собой яму абсолютно круглой формы. Путь является безопасным, если он не проходит по ловушке (но может проходить по ее границе).

Формат входных данных

Во входном файле записаны сначала координаты домика Винни-Пуха x_a, y_a , затем — координаты домика Кролика x_b, y_b , а затем — координаты центра и радиус ловушки x_c, y_c, r_c . Все координаты — целые числа из диапазона от -32000 до 32000. Радиус ловушки — натуральное число, не превышающее 32000.

Домики Винни-Пуха и Кролика не могут находиться внутри ловушки, но могут находиться на ее границе.

Формат выходных данных

Выведите в выходной файл одно число — длину самого короткого безопасного пути от домика Винни-Пуха до домика Кролика с точностью не менее 4 знаков после запятой.

Примеры

стандартный ввод	стандартный вывод
0 0 10 0 5 5 1	10
3 4 4 4 0 0 5	1

Задача Е. Целочисленные точки

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

Многоугольник (не обязательно выпуклый) на плоскости задан координатами своих вершин. Требуется подсчитать количество точек с целочисленными координатами, лежащих внутри него (но не на его границе).

Формат входных данных

В первой строке вводится n ($1 \leq n \leq 1000$) — число вершин многоугольника. В последующих n строках идут координаты (x_i, y_i) вершин многоугольника в порядке обхода по часовой стрелке. x_i и y_i — целые числа, по модулю не превосходящие 10^6 .

Формат выходных данных

Выведите одно число — количество искомых точек.

Пример

стандартный ввод	стандартный вывод
4 1 1 1 2 2 2 2 1	0

Задача F. Велорикши в Цзиньджене

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

В городе Цзиньджен есть N велорикш, каждый из которых дежурит в определённой точке и знает наизусть некоторый маршрут. К сожалению, не зная местного языка, вы не можете просить велорикшу остановиться на полпути, так что вынуждены доезжать до конца его маршрута, если прибегнули к его услугам.

Кроме того, вы можете передвигаться по городу между любыми двумя точками — для упрощения модели будем считать, что время передвижения равняется евклидовому расстоянию между точками начала и конца движения.

За какое наименьшее время можно добраться из точки A в точку B ?

Формат входных данных

В первой строке содержатся целые числа x_A, y_A — координаты точки, из которой вы начинаете движение, во второй строке — целые числа x_B, y_B — координаты точки, в которую вы хотите добраться. В третьей строке целое число n ($1 \leq n \leq 100$) — количество велорикш. Далее n строк по пять целых чисел: $x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2}, t$ — координаты начала и конца маршрута велорикши и время проезда по маршруту ($0 < t \leq 1500$).

Все координаты принадлежат интервалу $[0, 1000]$.

Формат выходных данных

Выведите одно число — минимальное время перемещения из точки A в точку B . Допустимая погрешность ответа — 10^{-4} .

Примеры

стандартный ввод	стандартный вывод
0 0 5 5 1 0 1 5 1 1	6.0
0 0 5 5 1 1 1 15 14 1	7.0710678118654755

Задача G. U2

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

Недавно Вася узнал, что через любые две точки на плоскости с различными x координатами можно провести одну и только одну параболу, уравнение которой будет иметь вид $y = x^2 + bx + c$, где b и c — действительные числа. Назовём такую параболу U -образной.

После этого он нарисовал на плоскости несколько различных точек с целыми координатами и через каждые две из них, имеющие различные координаты x , провёл U -образную параболу. Рисунок получился откровенно плохой, но Вася не теряет надежды найти число различных получившихся парабол, во внутренней области каждой из которых нет ни одной нарисованной точки. Помогите Васе.

Внутренней областью U -образной параболы в данной задаче называется часть плоскости, лежащая строго выше параболы, при этом ось y направлена вверх.

Формат входных данных

В первой строке находится единственное целое число n ($1 \leq n \leq 100\,000$) — число точек.

В следующих n строках находятся описания точек, в i -й из них находятся 2 целых числа x_i и y_i — координаты i -й точки. Гарантируется, что все точки различные, а так же что координаты по модулю не превосходят 10^6 .

Формат выходных данных

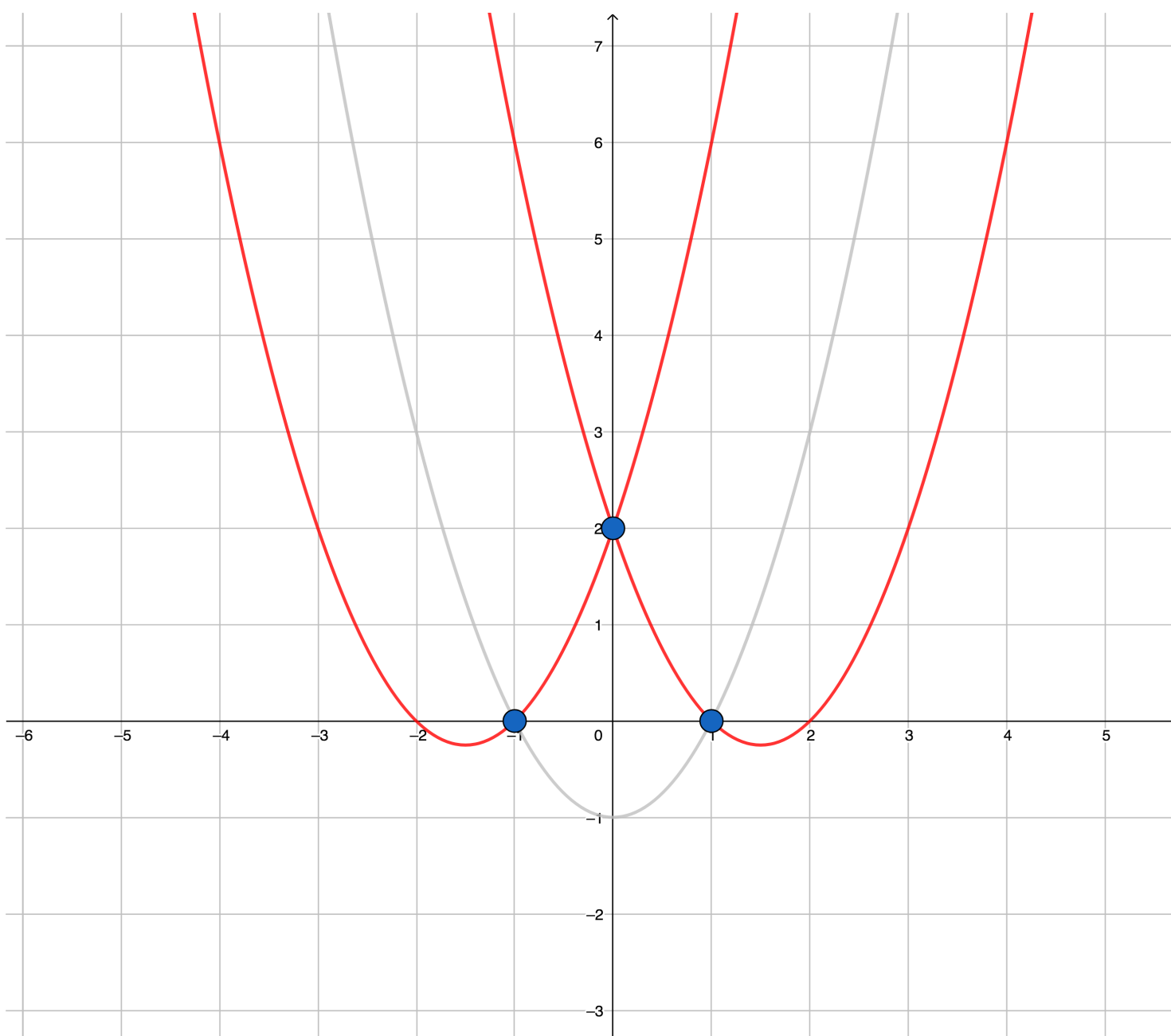
В единственной строке выведите единственное число — количество U -образных парабол, проходящих хотя бы через 2 точки и не содержащих никаких других точек в своей внутренней области (не считая границы).

Примеры

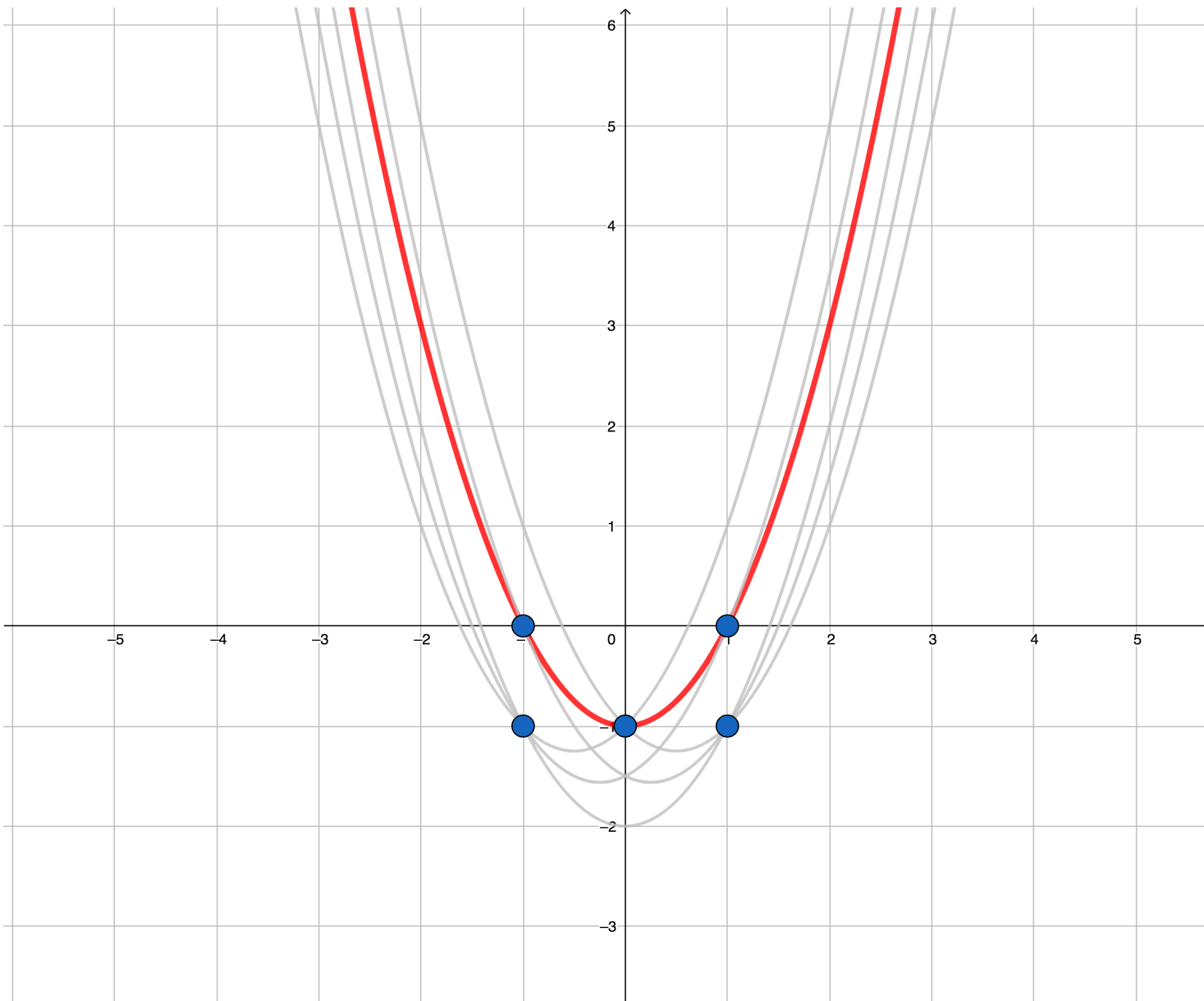
стандартный ввод	стандартный вывод
3 -1 0 0 2 1 0	2
5 1 0 1 -1 0 -1 -1 0 -1 -1	1

Замечание

На двух картинках ниже нарисованы все U -образные параболы, проходящие хотя-бы через 2 отмеченные точки в каждом из двух примеров. Красным отмечены те U -образные параболы, во внутренней области которых нет ни одной точки.



Первый пример.



Второй пример.

Задача Н. Извилистая ломаная

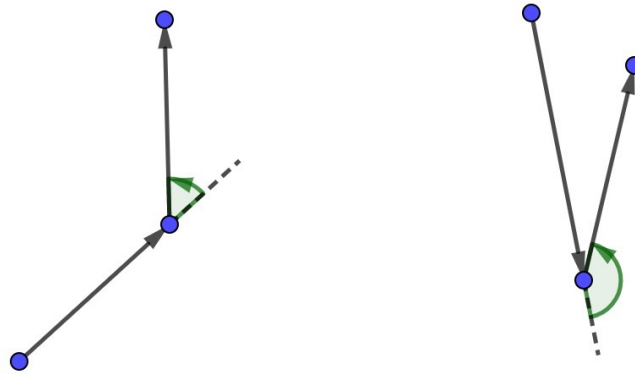
Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

У Васи есть n различных точек A_1, A_2, \dots, A_n на плоскости. Никакие три из них не лежат на одной прямой. Он хочет расположить их в некотором порядке $A_{p_1}, A_{p_2}, \dots, A_{p_n}$, где p_1, p_2, \dots, p_n — это некоторая перестановка чисел от 1 до n .

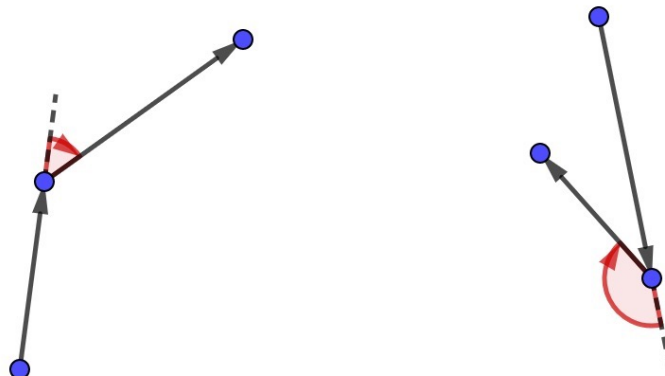
Сделав так, он нарисует ориентированную ломаную на этих вершинах, проведя направленные отрезки из каждой точки в следующую в выбранном порядке точку. То есть для всех $1 \leq i \leq n-1$ он проведет направленный отрезок из точки A_{p_i} в точку $A_{p_{i+1}}$. Он хочет, чтобы получившаяся ломаная удовлетворяла 2-м условиям:

- она будет несамопересекающейся, то есть любые 2 отрезка, которые не являются соседними, не имеют общих точек.
- она будет извилистой.

У Васи есть строка s , состоящая из $(n-2)$ -х символов “L” или “R”. Будем называть направленную ломаную извилистой, если её i -й поворот будет налево, если $s_i = \text{“L”}$ и направо, если $s_i = \text{“R”}$. Более формально: i -й поворот ломаной будет в точке $A_{p_{i+1}}$, в ней направленный отрезок из точки A_{p_i} в точку $A_{p_{i+1}}$ поменяется на направленный отрезок из точки $A_{p_{i+1}}$ в точку $A_{p_{i+2}}$. Обозначим вектор $\vec{v}_1 = \overrightarrow{A_{p_i}A_{p_{i+1}}}$ и вектор $\vec{v}_2 = \overrightarrow{A_{p_{i+1}}A_{p_{i+2}}}$. Тогда если для того, чтобы повернуть вектор \vec{v}_1 на наименьший возможный угол, чтобы его направление совпало с направлением вектора \vec{v}_2 надо сделать поворот против часовой стрелки, то будем говорить, что i -й поворот налево, а иначе направо. Для лучшего понимания посмотрите картинку, на которых изображены различные варианты поворотов:



На этой картинке изображены повороты налево



На этой картинке изображены повороты направо

Вам даны координаты точек A_1, A_2, \dots, A_n на плоскости и строка s . Найдите перестановку p_1, p_2, \dots, p_n чисел от 1 до n , такую что ломаная, которую нарисует Вася, будет удовлетворять двум заданным условиям.

Формат входных данных

В первой строке написано одно целое число n — количество точек ($3 \leq n \leq 2000$). В следующих n строках написаны по два целых числа x_i и y_i , разделённые пробелом — координаты точки A_i на плоскости ($-10^9 \leq x_i, y_i \leq 10^9$). В последней строке написана строка s из символов “L” и “R” длины $(n - 2)$. Гарантируется, что все точки различны и никакие три точки не лежат на одной прямой.

Формат выходных данных

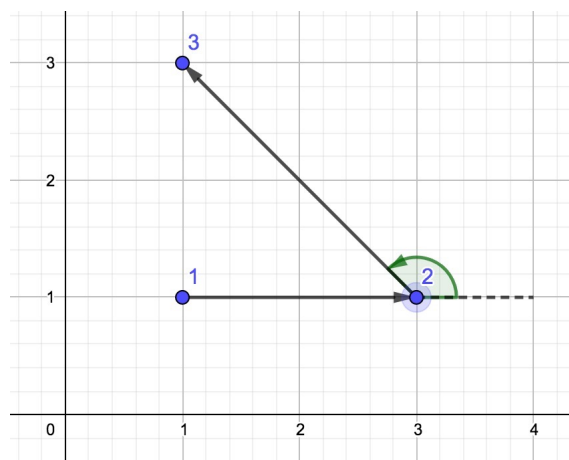
Если подходящей перестановки не существует выведите -1 . Иначе выведите n чисел p_1, p_2, \dots, p_n — найденную перестановку ($1 \leq p_i \leq n$ и все p_1, p_2, \dots, p_n различны). Если подходящих перестановок несколько, выведите любую.

Примеры

стандартный ввод	стандартный вывод
3 1 1 3 1 1 3 L	1 2 3
6 1 0 0 1 0 2 -1 0 -1 -1 2 1 RLLR	5 4 1 6 2 3

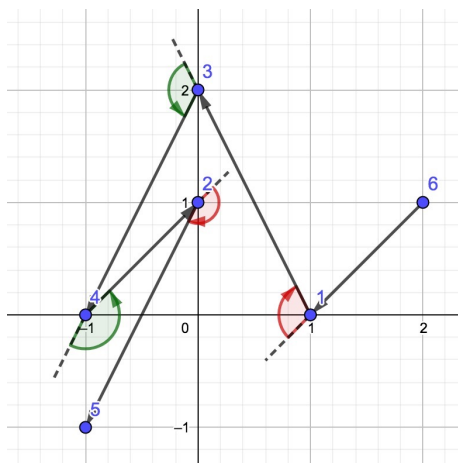
Замечание

Вот картинка, изображающая ломаную из 1 теста:



Как мы видим, эта ломаная несамопересекающаяся, а также извилистая, так как поворот в точке 2 налево.

Вот картинка, изображающая ломаную из 2 теста:



Задача I. Brainfuck

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

Мы продолжаем курс программирования на языке brainfuck <https://ru.wikipedia.org/wiki/Brainfuck>. Напоминаем, что отличием нашей версии от обычного языка будет только то, что количество ячеек памяти неограниченно. Обратите внимание, что при работе ваша программа не должна делать больше $5 \cdot 10^7$ действий, а так же длина исходной программы не должна превышать $5 \cdot 10^5$ символов.

В нашем втором уроке языка brainfuck мы научимся циклам и массивам, а именно, вам будет предложено отсортировать массив.

От вас требуется написать программу, **генерирующую корректную программу на языке brainfuck**, и выводящую её в стандартный поток вывода. Формат ввода и вывода, с которым должна работать ваша сгенерированная программа, описан ниже. Обратите внимание, что ваша сгенерированная программа не должна содержать символов перевода строки.

Для удобства вот ссылка на тестировщик программ на языке brainfuck <http://algocode.ru/docs/docs/bff.cpp>.

Формат входных данных

Сгенерированной вами программе на стандартный поток ввода в первой строке будет передаваться число n ($2 \leq n \leq 15$) — размер массива.

В следующих n строках вам будет передано по одному числу a_i ($0 \leq a_i < 256$) — значения элементов массива. Все числа разделены знаком переноса строки.

Формат выходных данных

Сгенерированная вами программа через пробел должна вывести n чисел — значения элементов массива в отсортированном по возрастанию порядке.

Система оценки

Всего 20 тестов, каждый стоит по 50 баллов, балл уменьшается в зависимости от числа итераций вашего решения. В первых 10 тестах все числа от 1 до 9

Пример

стандартный ввод	стандартный вывод
3	1 2 3
3	
1	
2	

Замечание

Подумайте, как удобнее всего хранить массив, чтобы по нему можно было удобно итерироваться. Для сортировки я рекомендую сортировку пузырьком, её проще всего писать. У вас наверное возникнут трудности в сравнении двух чисел. Подумайте, как использовать для этого предыдущую задачу.