

Задача 1. Головоломка

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Миша и Паша разрабатывают компьютерную игру, для которой Миша придумал следующую головоломку.

Дан циферблат с некоторым количеством делений. Деления занумерованы по часовой стрелке числами, начиная с 1. В центре циферблата прикреплены две стрелки. Каждая стрелка указывает на какое-то деление.

Если обе стрелки указывают на одно и то же деление, то число на этом делении загорается. Для решения головоломки требуется, чтобы все числа загорелись.

Если стрелки изначально указывают на одно и то же деление, то число на этом делении сразу загорается.

За один ход можно сделать одно из двух действий:

1. Передвинуть первую стрелку по часовой или против часовой стрелки на одно деление.
2. Передвинуть обе стрелки на одно деление. В этом случае обе стрелки передвигаются в одном направлении (либо обе по часовой стрелке, либо обе — против).

Паша понял, что решение этой головоломки не составляет труда, и поэтому предложил сделать усложненную версию. В ней изменено второе действие: вместо движения обеих стрелок в одном направлении, стрелки двигаются в разные стороны. Стрелки так же двигаются за одно действие на одно деление, если одна из них по часовой, то другая — против, и наоборот. Первое действие остается таким же, как у Миши.

Для того, чтобы оценить сложность загадки, ребята хотят понять, какое минимальное количество действий нужно, чтобы решить Пашину или Мишину версию головоломки.

Формат входных данных

Первая строка входного файла содержит три целых числа N, f, s — количество делений на циферблате, номер деления, на которое указывает первая стрелка, и номер деления на которое указывает вторая стрелка соответственно ($1 \leq N \leq 10^6, 1 \leq f, s \leq N$).

Во второй строке написано число t , указывающее для чьей версии головоломки нужно дать ответ ($1 \leq t \leq 2$): 1 — для Мишиной, а 2 — для Пашиной.

Формат выходных данных

В выходной файл необходимо вывести одно целое число — минимальное количество действий для решения указанной головоломки.

Примеры

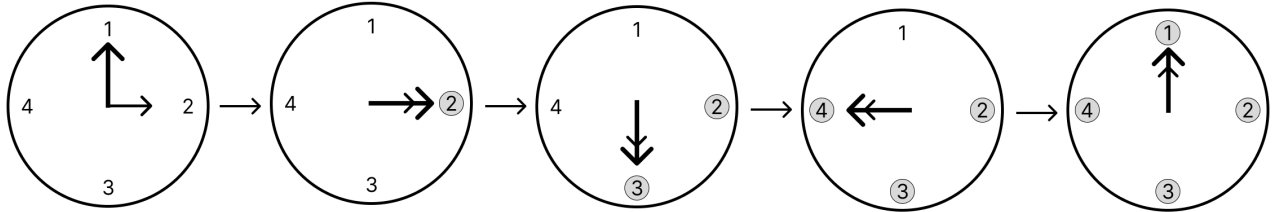
input.txt	output.txt
4 1 2 1	4
4 4 2 2	8
2 1 1 1	1

Пояснение к примеру

В первом примере для решения головоломки за 4 действия можно первым действием

подвинуть первую стрелку по часовой стрелке, а оставшимися тремя действиями двигать обе стрелки по часовой.

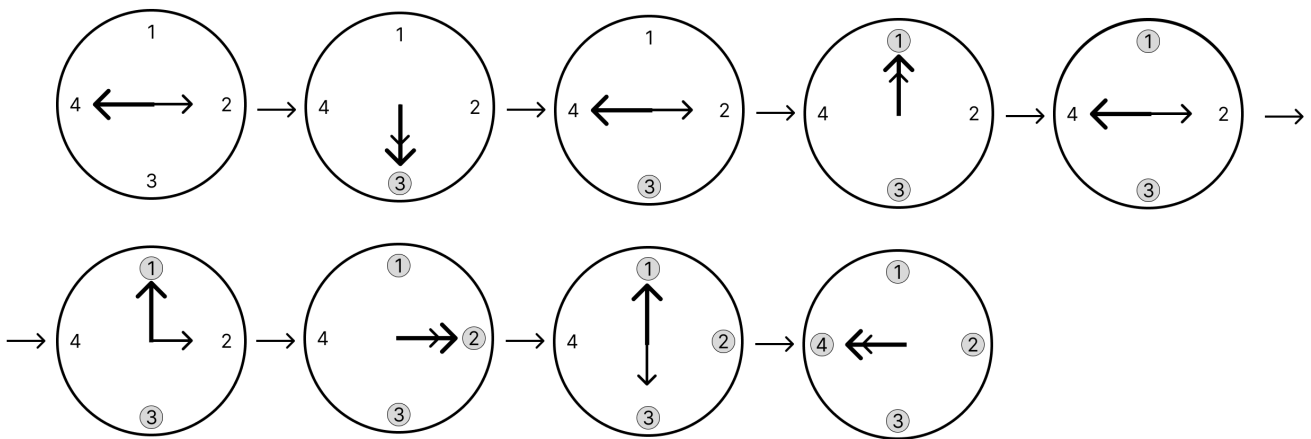
На рисунке ниже изображено состояние головоломки после каждого действия. Большая стрелка — это первая стрелка, маленькая — вторая.



Во втором примере возможная подходящая последовательность действий представлена ниже:

1. поворот первой стрелки против часовой, второй — по часовой;
2. поворот первой стрелки по часовой, второй — против часовой;
3. поворот первой стрелки по часовой, второй — против часовой;
4. поворот первой стрелки против часовой, второй — по часовой;
5. поворот первой стрелки по часовой;
6. поворот первой стрелки по часовой;
7. поворот первой стрелки против часовой, второй — по часовой;
8. поворот первой стрелки против часовой, второй — по часовой.

Так же, как и для первого примера, на рисунке ниже изображено состояние головоломки после каждого действия.



Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	0	Тесты из условия	
2	40	$t = 1$	1
3	60	$t = 2$	1

Задача 2. Крутилки и вертелки

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Сегодня на уроке труда Петя вместе с одноклассниками вытачивал из дерева шестеренки самых разных размеров.

Больше всего Пете понравилось соединять шестеренки в большие конструкции, начинать крутить **одну** произвольную шестеренку и смотреть, как вся конструкция начинала вращаться. Но вот незадача, очередная собранная школьником конструкция отказывалась вращаться. Кажется, Петя уже начал догадываться, в чем причина.

Шестеренку на плане можно представить окружностью радиуса r с центром в точке (x, y) . Считаем, что две шестеренки соединены, если соответствующие им окружности имеют только **одну** общую точку. Если две шестеренки соединены, и одна из них вращается **по** часовой стрелке, то вторая будет вращаться **против** часовой стрелки. Гарантируется, что две окружности имеют **не более одной** общей точки, и никакая окружность не лежит внутри другой.

Школьник просит вас помочь определить, можно ли начать крутить **одну** шестеренку так, чтобы **все** шестеренки в собранной конструкции начали вращаться.

Формат входных данных

В первой строке входного файла дано количество тестов T ($1 \leq T \leq 30$).

Далее идут T блоков с описанием тестов. Первая строка блока содержит одно целое число N — количество шестеренок ($1 \leq N \leq 5000$).

Далее в блоке следует N строк, i -я строка содержит три целых числа x_i, y_i, r_i — координаты i -й шестеренки на плоскости и ее радиус ($-10^9 \leq x_i, y_i \leq 10^9, 1 \leq r_i \leq 10^9$).

Формат выходных данных

В выходной файл для каждого теста требуется вывести в отдельной строке слово YES, если все шестеренки в конструкции, собранной Петей, могут вращаться. В противном случае выведите слово NO.

Система оценки

Баллы начисляются за каждый пройденный тест. Баллы за тесты из условия не начисляются.

Примеры

input.txt	output.txt
1 4 0 0 4 0 8 4 8 0 4 8 8 4	YES
2 3 0 0 1 3 0 2 0 4 3 3 1 1 1 1 4 2 1 9 3	NO YES

Задача 3. VR игра

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Тимофей любит программирование, а еще любит играть в компьютерные игры. У него есть мечта — создать свою компьютерную игру, в которую будет играть много других людей.

Недавно Тимофею подарили VR (Virtual Reality) гарнитуру, и он понял, что хочет написать игру именно для VR. У него уже есть идея: игрок находится в некоторой точке, а перед ним в воздухе висят кольца различного диаметра. Игроку необходимо бросить снаряд (для простоты размером с материальную точку), который будет лететь по параболической траектории из его начальной позиции, в заданную конечную позицию. Задача игрока — бросить снаряд таким образом, чтобы он попал в наибольшее количество колец.

Сейчас игра на этапе проектирования и тестирования её различных частей. Для упрощения первоначальных расчётов Тимофей спроецировал игру на двумерную плоскость, разместил начальную позицию игрока в точке $(0, 0)$, конечную позицию в некоторой точке $(X, 0)$, а кольца представил отрезками, всегда перпендикулярными оси абсцисс. Тимофею хочется понять, каким образом оценивать результат игроков, и для этого он решил написать программу, которая определяет по заданным начальным и конечным координатам, а также расположению колец, какой максимальный результат может получить игрок. Помогите ему написать такую программу.

Формат входных данных

В первой строке входного файла даны два целых числа X и N , где X — координата конечной точки, куда попадет снаряд, N — количество колец ($2 \leq X \leq 10^6$, $0 \leq N \leq 2 \cdot 10^5$).

В каждой из следующих N строк заданы четыре целых числа x_{i1} , y_{i1} , x_{i2} , y_{i2} — координаты начала и конца отрезка, являющегося проекцией кольца ($1 \leq x_{i1} = x_{i2} < X$, $1 \leq y_{i1} < y_{i2} \leq 10^6$, $1 \leq i \leq N$).

Гарантируется, что никакие два отрезка не пересекаются.

Формат выходных данных

В первую строку выходного файла необходимо вывести ответ на задачу — максимальное количество колец, которое может пересечь снаряд. Во второй строке необходимо в любом порядке вывести через пробел номера колец, которые пересечет снаряд.

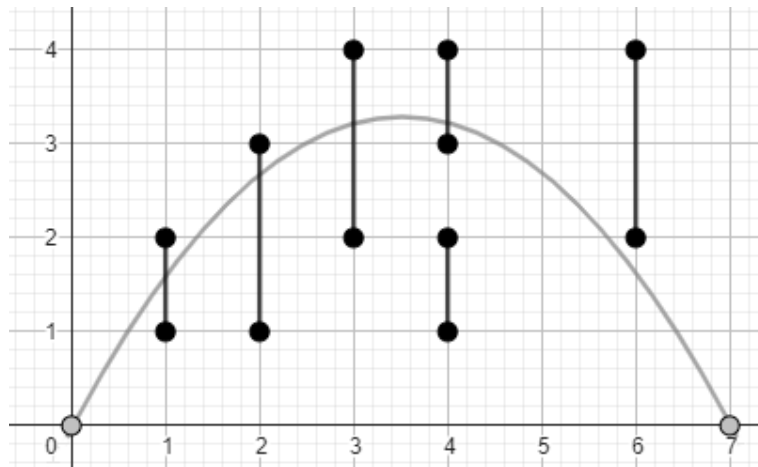
Пример

input.txt	output.txt
7 6	4
1 1 1 2	1 2 3 5
2 1 2 3	
3 2 3 4	
4 1 4 2	
4 3 4 4	
6 2 6 4	

Пояснение к примеру

Уравнение параболы, проходящей через точки $(x_0, 0)$ и $(x_1, 0)$ можно описать как $y = a \cdot (x - x_0) \cdot (x - x_1)$.

Ниже представлен рисунок проекции из первого тестового примера, показывающий один из вариантов броска снаряда.



Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	0	Тесты из условия	
2	20	$N \leq 100, y_i \leq 10$	1
3	30	$N \leq 5000$	1, 2
4	50	$N \leq 2 \cdot 10^5$	1, 2, 3

Задача 4. Белуга и Скиттл

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт



Белуга решил нарисовать своему другу Скиттлу чёрно-белый пиксель-арт. Белуга не был бы собой, если бы не стал шутить — он собирается спрятать в своей работе как можно больше «амогусов». Амогус — небольшое изображение размером $A \times B$, которое должно находиться внутри работы — полотна размера $H \times W$, которое Белуга собирается отправить. Рисовать он их научился, а вот считать — нет. Помогите Белуге определить, сколько изображений амогусов можно встретить в картине.

Формат входных данных

В первой строке входного файла записаны четыре целых числа: A , B , H , W — размеры амогуса и размеры работы ($2 \leq A \leq H \leq 2000$, $2 \leq B \leq W \leq 2000$).

В следующих A строках задано описание амогуса в виде строк длины B , состоящих из символов '.' и '#'.
В следующих H строках в таком же формате задан сам рисунок.

Формат выходных данных

В выходной файл выведите единственное число — сколько раз изображение амогуса встречается в рисунке (пересекающиеся вхождения допустимы).

Примеры

input.txt	output.txt
<pre>4 4 4 4 .### ##.. #### .#.# .### ##.. #### .#.#</pre>	1
<pre>4 4 9 9 .### ##.. #### .#.# .###.### ##...##.. ####.#### .#.#...#.####.### ##...##.. ####.#### .#.#...#.#</pre>	4

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	0	Тесты из условия	
2	10	$A = B = 4, H, W \leq 10$	1
3	30	$A = B = 4, H, W \leq 2000$	1, 2
4	20	$A, B, H, W \leq 100$ амогус может быть произвольного вида	1, 2
5	40	$A, B, H, W \leq 2000,$ амогус может быть произвольного вида	1, 2, 3, 4

Замечание

Во второй и третьей подзадачах вид амогуса такой же, как в примерах.

Задача 5. Арсений и удаление файлов

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Арсений принялся за удаление ненужных файлов на своем ноутбуке. Файловый менеджер, которым он пользуется, отображает файлы и папки, а также позволяет выделять некоторые из них. При этом в интерфейсе менеджера напротив каждого объекта отображается чекбокс, который может находиться в одном из трех состояний: , , .

Опишем правила, по которым проставляются эти чекбоксы.

1. Если объект является файлом:
 - а) если он выделен, напротив него ставится
 - б) иначе — ставится .
2. Если объект является папкой:
 - а) если все объекты в папке помечены , напротив нее ставится
 - б) если все объекты в папке помечены , напротив нее ставится
 - в) иначе — ставится .

Изначально ни один объект не выделен (напротив всех объектов стоит .

Отметим, что внутри папок могут находиться не только файлы, но и другие папки. То есть для того, чтобы определить состояние чекбокса внешней папки, нужно знать состояния чекбоксов внутренних папок.

При нажатии на чекбокс его состояние изменяется:

- состояния и при нажатии переходят в
- состояние переходит в .

При этом состояния чекбоксов связанных дочерних и родительских объектов тоже могут измениться.

Рассмотрим, например, такую файловую структуру:

- Папка 1
 - Папка 2
 - Файл 1
 - Файл 2
 - Папка 3
 - Файл 3
 - Файл 4

Если мы нажмем на чекбокс возле «Папки 3», то состояния изменятся следующим образом:

- Папка 1
 - Папка 2
 - Файл 1
 - Файл 2
 - Папка 3
 - Файл 3
 - Файл 4

Арсений выделяет объекты, которые он хочет удалить, или снимает выделение с объектов, которые он хочет оставить. В любой момент он хочет уметь узнавать состояние чекбокса любого объекта. Помогите ему ответить на его запросы.

Формат входных данных

В первой строке входного файла записано одно целое число N — количество объектов на ноутбуке Арсения ($1 \leq N \leq 2 \cdot 10^5$). Для удобства все объекты занумерованы от 1 до N .

В $i+1$ -й строке следует описание i -го объекта. Первым записано целое число M_i — количество его дочерних объектов. Если $M_i = 0$, значит i -й объект является файлом (**на ноутбуке Арсения нет пустых папок**). Далее через пробел следуют M_i целых чисел: номера дочерних объектов.

В $N + 2$ -й строке записано число целое число Q — количество запросов ($1 \leq Q \leq 10^5$).

Далее следуют Q строк с описаниями запросов. Запросы могут быть двух видов:

- ! i — нажать на чекбокс напротив i -го объекта;
- ? i — узнать состояние чекбокса i -го объекта.

Формат выходных данных

Для каждого запроса 2-го вида (? i) нужно вывести в выходной файл ответ — код состояния чекбокса i -го объекта на момент запроса.

Состояния кодируются следующим образом: — 0, — 1, — 2.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	0	Тесты из условия	
2	32	$N, Q \leq 1000$	1
3	16	Внутри папок находятся только файлы	
4	52	Нет дополнительных ограничений	1, 2, 3

Примеры

input.txt	output.txt
7	2
2 2 5	1
2 3 4	2
0	2
0	0
2 6 7	1
0	1
0	
10	
! 5	
? 6	
? 1	
! 1	
? 1	
? 2	
! 4	
? 4	
? 2	
? 1	
2	0
0	2
0	0
7	2
? 1	
! 1	
? 1	
! 1	
? 1	
! 2	
? 2	

Задача 6. Две фотографии

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Иннокентий разбирал свои архивы и наткнулся на две очень похожие фотографии холмистой местности. К сожалению, он не смог вспомнить, где они были сделаны. Несмотря на это, Иннокентий хочет понять, могло ли одно и то же место быть запечатлено на этих фотографиях.

Пусть ландшафты представлены в виде ломаных $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ и $(X_1, Y_1), (X_2, Y_2), \dots, (X_M, Y_M)$ размера N и M соответственно. Тогда Иннокентий считает их одинаковыми, если после масштабирования второй ломаной по обоим осям её можно целиком наложить на какую-то часть первой ломаной.

Одну ломаную можно наложить на другую, если существует вектор, при перемещении на который одной из ломаных, все ее вершины совпадут с соответствующими вершинами второй.

Формат входных данных

В первой строке входного файла дано количество тестов T ($1 \leq T \leq 10^5$).

Далее идут T блоков с описанием тестов.

В первой строке блока заданы длины ломаных N и M ($3 \leq M \leq N \leq 2 \cdot 10^5$).

Во второй строке блока даны $2 \cdot N$ чисел — описание первой ломаной: $x_1, y_1, \dots, x_N, y_N$ ($|x_i|, |y_i| \leq 10^9$).

В третьей строке блока даны $2 \cdot M$ чисел — описание второй ломаной: $x_1, y_1, \dots, x_M, y_M$ ($|x_i|, |y_i| \leq 10^9$).

Гарантируется, что $x_i < x_{i+1}$ и $y_i \neq y_{i+1}$ для любой ломаной, и что сумма N по всем тестам не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого теста в отдельную строку в выходной файл нужно вывести слово YES, если вторую ломаную можно наложить на первую. В противном случае нужно вывести слово NO.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	0	Тесты из условия	
2	21	$\sum_{i=1}^t N_i \leq 5000$	1
3	20	$\sum_{i=1}^t N_i \leq 2 \cdot 10^5,$ $x_{i+1} = x_i + 1, y_{i+1} - y_i = 1$	1
4	23	$\sum_{i=1}^t N_i \leq 2 \cdot 10^5, x_{i+1} = x_i + 1$	1, 3
5	36	Без дополнительных ограничений	1, 2, 3, 4

Пример

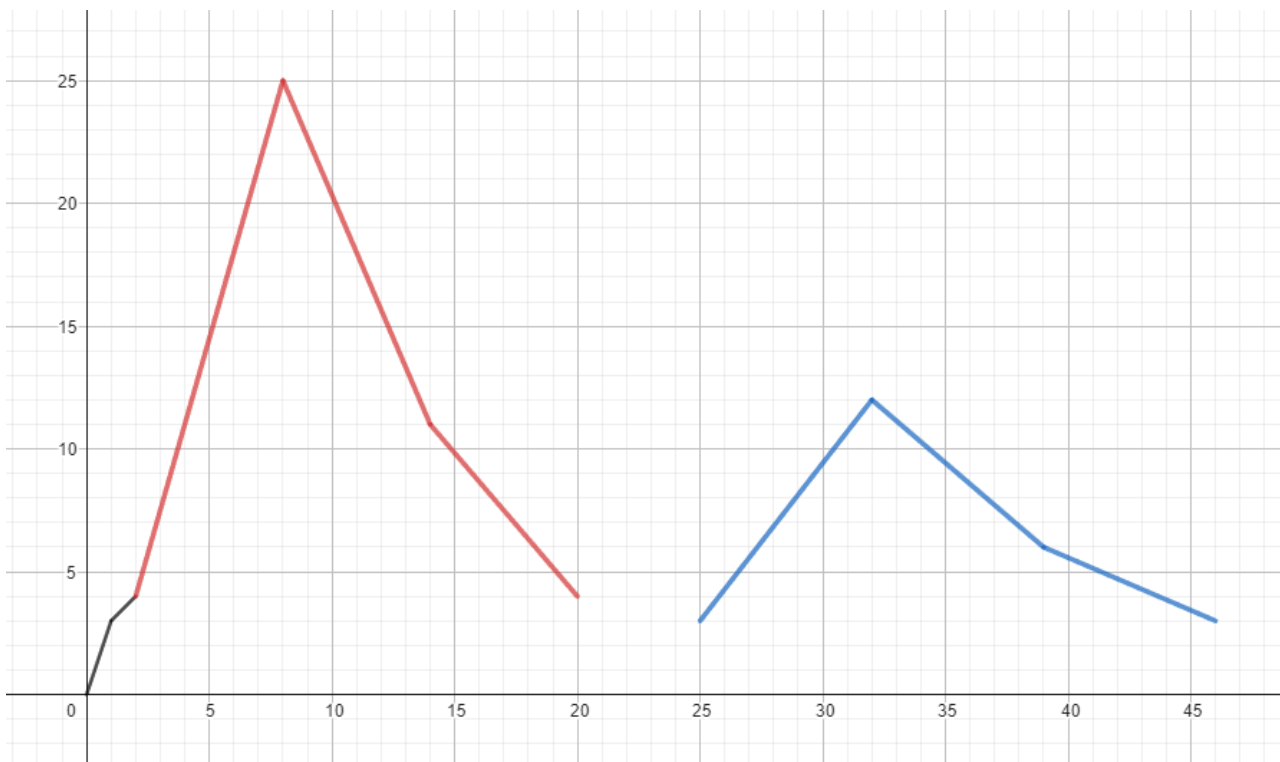
input.txt	output.txt
4	YES
3 3	NO
0 0 1 2 2 0	NO
0 0 10 2 20 0	YES
3 3	
0 0 1 -2 2 0	
0 0 10 2 20 0	
3 3	
0 0 1 2 2 0	
0 0 1 2 3 0	
6 4	
0 0 1 3 2 4 8 25 14 11 20 4	
25 3 32 12 39 6 46 3	

Пояснение к примеру

В первом примере можно сжать в 10 раз ломаную по оси абсцисс, после чего вторая ломаная наложится на первую.

Во втором примере вторую ломаную наложить на первую не получится, так как отразить ломаную вдоль какой-либо из осей нельзя.

В третьем примере можно показать, что не существует преобразований второй ломаной, позволяющих наложить ее на первую.



В четвертом примере вторую ломаную, изображенную на рисунке справа, можно наложить на часть первой ломаной (левый рисунок), если растянуть вторую ломаную вдоль оси абсцисс в 2.(3) раза и сжать её по оси ординат примерно в 0.8571429 раз.

Задача 7. Спецагент Петя

Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1,5 секунды
Ограничение по памяти:	256 мегабайт

Спецагенту Пете предоставили новое устройство — крюк-кошку, с помощью которого он сможет зацепляться за разные предметы во время выполнения заданий. Но сначала нужно научиться им пользоваться. Для этого он отправляется на полигон, где хочет переместиться из одной точки в другую. Для этого специальная платформа метнёт его в выбранном направлении. Во время полета Петя может зацепляться за специальные столбы и корректировать свой полет так, чтобы приземлиться в точке назначения и при этом не столкнуться с препятствием.

Пете важен стиль. Поэтому Петя хочет добраться до финиша, преодолев наименьшее расстояние, и при этом он будет зацепляться за столбы только если:

- Петя находится на расстоянии R от столба;
- в любой другой точке полёта он находится на расстоянии, большем R .

Более формально, Петя может зацепиться за столб только в точке касания окружности радиуса R с центром в столбе и продолжить своё движение по ней до момента отцепления.

Поскольку Петя хочет научиться пользоваться крюком, **он должен воспользоваться им хотя бы один раз.**

Формат входных данных

В первой строке входного файла даны три целых числа: N , M и R — количество столбов, количество преград и расстояние, на котором Петя будет использовать свой крюк ($1 \leq N \leq 200$, $0 \leq M \leq 200$, $1 \leq R \leq 50$).

В следующих двух строках даны координаты x и y начальной и конечной точек ($0 \leq |x|, |y| \leq 10^6$).

В следующих N строках даны координаты столбов x и y ($0 \leq |x|, |y| \leq 10^6$). Гарантируется, что расстояние между любыми двумя столбами не меньше $2 \cdot R$, и что расстояние от начальной и конечной точек до столбов не меньше R .

В следующих M строках даны координаты концов препятствий x_1, y_1, x_2, y_2 ($0 \leq |x_1|, |x_2|, |y_1|, |y_2| \leq 10^6$).

Формат выходных данных

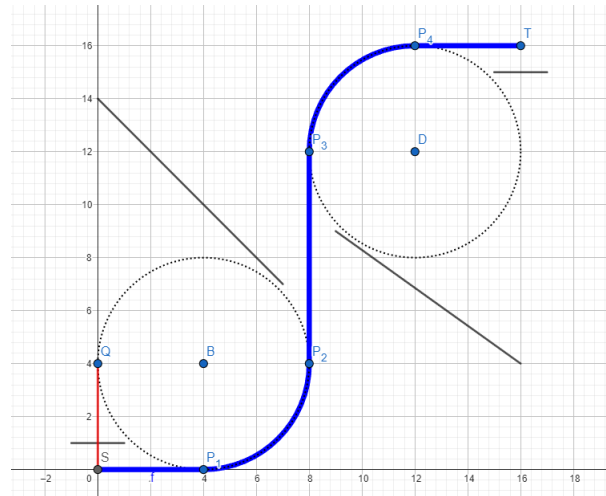
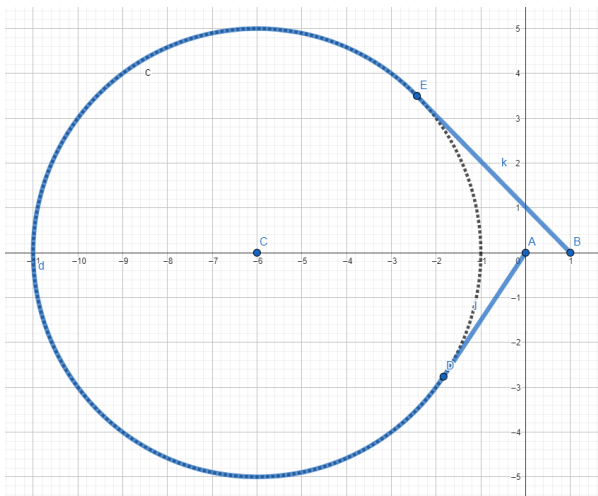
В выходной файл требуется вывести одно число — минимальную длину пути от начальной точки до конечной, или **-1**, если пути не существует.

Ваш ответ будет засчитан, если его абсолютная или относительная ошибка не будет превосходить 10^{-6} — т.е., формально, если $\min(|x - y|, \frac{|x-y|}{x}) \leq 10^{-6}$, где x — ответ жюри, а y — ответ, который выдаёт ваша программа.

Примеры

input.txt	output.txt
<pre>1 0 5 0 0 1 0 -6 0</pre>	32.82713622798212671494
<pre>2 4 4 0 0 16 16 4 4 12 12 -1 1 1 1 15 15 17 15 0 14 7 7 9 9 16 4</pre>	28.566370614359171113172
<pre>1 0 3 0 0 6 0 3 2</pre>	6.36874671819856885691

Пояснение к примеру



В первом примере один из возможных путей Пети отмечен жирной линией.

Во втором примере Петя начинает в точке **A** и хочет закончить в точке **B**. В процессе полета он может зацепиться за столбы **C** и **D** в точках касания к окружностям, отмеченных пунктирными линиями. Единственный возможный путь отмечен жирной линией, остальные — пересекают препятствия в некоторых точках.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Всесибирская открытая олимпиада школьников по информатике
Заключительный этап, 9-11 классы, 25 февраля 2024 г.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	0	Тесты из условия	
2	18	$M = 0$	
3	19	$N \leq 10, 1 \leq M \leq 10$, расстояние от препятствий до столбов больше R	1
4	12	$N \leq 50, 1 \leq M \leq 50$, расстояние от препятствий до столбов больше R	1, 3
5	21	$N \leq 50, 1 \leq M \leq 50$	1, 3, 4
6	30	Без дополнительных ограничений	1, 2, 3, 4, 5