

Задача 1. Вкусняшки

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

Антон всегда очень любил животных. Счастью не было предела, когда родители подарили ему щенка по кличке Бобик.

Антон довольно быстро придумал для своего питомца увлекательную игру. Он разбрасывал по комнате различные лакомства (вкусняшки), располагал Бобика в центре комнаты и наблюдал, куда же побежит щенок. Вскоре Антон заметил, что Бобик всегда бежит к той вкусняшке, которая лежит ближе всего к нему. Если таких лакомств несколько, то Бобик бежал к любому из ближайших. Добравшись до вкусняшки, щенок радостно съедал ее и бежал к следующему лакомству по тому же принципу. Съев определенное количество вкусняшек, Бобик останавливался в месте, где лежало последнее съеденное лакомство.

Помогите Антону узнать, какие вкусняшки съест Бобик, если известно расположение всех вкусняшек, начальное положение самого щенка, а также сколько вкусняшек он съест.

Для простоты будем считать, что вкусняшки разбросаны вдоль одной прямой. Антон усаживает Бобика в некоторую точку на этой прямой и проводит свой эксперимент.

Формат входных данных

В первой строке входного файла задано три целых числа N , M и X , где N — количество вкусняшек, разбросанных на прямой, M — количество вкусняшек, которые съест Бобик, X — положение Бобика ($1 \leq N \leq 10^5$, $1 \leq M \leq N$, $-10^9 \leq X \leq 10^9$).

В следующей строке записано N различных целых чисел $x_1, x_2 \dots x_N$, где x_i — координата i -й вкусняшки ($-10^9 \leq x_i \leq 10^9$, $1 \leq i \leq N$).

Формат выходных данных

В выходной файл необходимо вывести в одну строку через пробел M целых чисел — координаты точек, в которых Бобик съест лакомства, в порядке их поедания. Если есть несколько вариантов поедания вкусняшек, можно вывести любой.

Примеры

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

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	$N \leq 1000$	1
3	60	Без дополнительных ограничений	1, 2

Задача 2. Преобразование карты

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

Иногда для визуализации решения бывает полезно преобразовать изображение точек на карте в табличное представление. Как известно, координаты на карте задаются широтой и долготой. Широта может принимать значения от -90 до 90 , а долгота — от -180 до 180 . Широта в таблице будет соответствовать строке, а долгота — столбцу. Максимальная широта будет располагаться на самой верхней строке таблицы, а максимальная долгота — в самом правом столбце.

Вам даны координаты городов. Необходимо их расположить на клеточном поле так, чтобы было красиво. Понятие красоты весьма субъективно, поэтому будем считать, что требуется расположить эти города как можно компактнее. При этом требуется сохранить их положение относительно друг друга. Это означает, что если один город на карте имел координату долготы больше, чем другой, то и в таблице он должен находиться правее (восточнее) этого города. Аналогично, если координата широты первого города больше координаты широты второго города, то и в таблице первый город должен располагаться выше (севернее) второго, как на географической карте. Кроме того, нужно добиться того, чтобы каждый город был в своей отдельной клетке.

Однако, хочется как-то ужать таблицу — минимизировать количество строк и столбцов в ней. Поэтому принято решение «округлить» координаты в соответствии с заданными параметрами h и w , где h — размер одной клетки в таблице по широте, а w — размер одной клетки таблицы по долготе.

Пусть город имеет координаты на карте (x, y) , причем $h \cdot k \leq x + 90 < h \cdot (k + 1)$ и $w \cdot r \leq y + 180 < w \cdot (r + 1)$, где x — широта, а y — долгота города, а k и r — некоторые неотрицательные целые числа. Тогда этот город будет помещен в клетку таблицы с координатами (k, r) . После того, как все города размещены в таблице, из нее нужно убрать пустые строки и столбцы.

Вам предстоит узнать, получится ли разместить города в отдельные клетки после «округления» координат. Если это удастся сделать, то при наличии соответствующего запроса нужно будет распечатать полученную таблицу.

Формат входных данных

В первой строке входного файла записано число N — количество городов ($1 \leq N \leq 60000$).

В следующих N строках записано по два целых числа — координаты городов (широта и долгота). Широта может принимать значения от -90 до 90 , а долгота — от -180 до 180 . Координаты у всех городов различные. Города нумеруются числами от 1 до N в порядке перечисления их координат.

Далее в последнюю строку через пробел записано три целых числа h , w и q , где h и w — параметры округления координат, а q — вид запроса ($1 \leq h \leq 180$, $1 \leq w \leq 360$, $1 \leq q \leq 2$).

Формат выходных данных

Если при заданных h и w разместить города в таблицу однозначным образом не удастся, то в первую строку выходного файла нужно вывести слово **No**, а во вторую строку нужно вывести номера двух городов, которые попали после округления координат в одну и ту же клетку. Если таких городов несколько, можно вывести любые.

Если таблицу построить удалось, тогда в первую строку выходного файла при любом значении q нужно вывести слово **Yes**, а во вторую строку нужно записать через пробел два целых числа a и b — количество строк и количество столбцов построенной таблицы.

Если q равно 2, то дальше необходимо напечатать саму таблицу. Это должно быть a строк, содержащих по b чисел, записанных через пробел — содержимое ячеек таблицы. Если ячейка таблицы пуста, то нужно вывести 0, иначе нужно выдать номер города, который размещен в ней.

Для того, чтобы сохранить соответствие таблицы карте, строки таблицы нужно выводить, начиная с верхней.

Примеры

input.txt	output.txt
4 2 6 8 3 7 7 6 0 3 2 2	Yes 2 3 4 2 3 0 0 1
4 2 6 8 3 7 7 6 0 4 4 1	No 2 4
4 -30 -30 30 -30 -30 60 30 60 40 40 2	Yes 2 2 2 4 1 3

Пояснение к примеру

На рисунке ниже в левой таблице отображено расположение городов из первого примера в соответствии с их координатами. Не трудно заметить, что каждый город попадает в свою ячейку размерности 3×2 . В правой таблице — расположение этих городов после «округления» их координат. Пустые строки и столбцы отмечены цветом, их нужно удалить.

	0	1	2	3	4	5	6	7
8				2				
7								3
6	4							
5								
4								
3								
2							1	
1								
0								

	90	91	92	93
32	4	2		3
31				
30				1

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	$N \leq 100, q = 1$	1
3	20	$N \leq 100, q = 2$	1, 2
4	30	$N \leq 30000,$ все координаты положительные	1, 2, 3
5	40	Без дополнительных ограничений	1, 2, 3, 4

Задача 3. 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

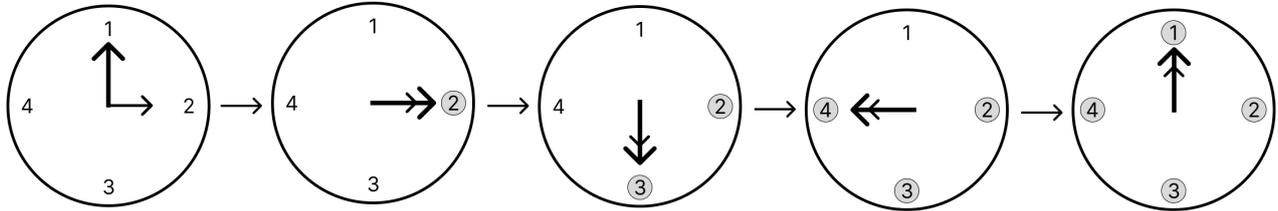
input.txt	output.txt
4 1 2 1	4
4 4 2 2	8
2 1 1 1	1

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.

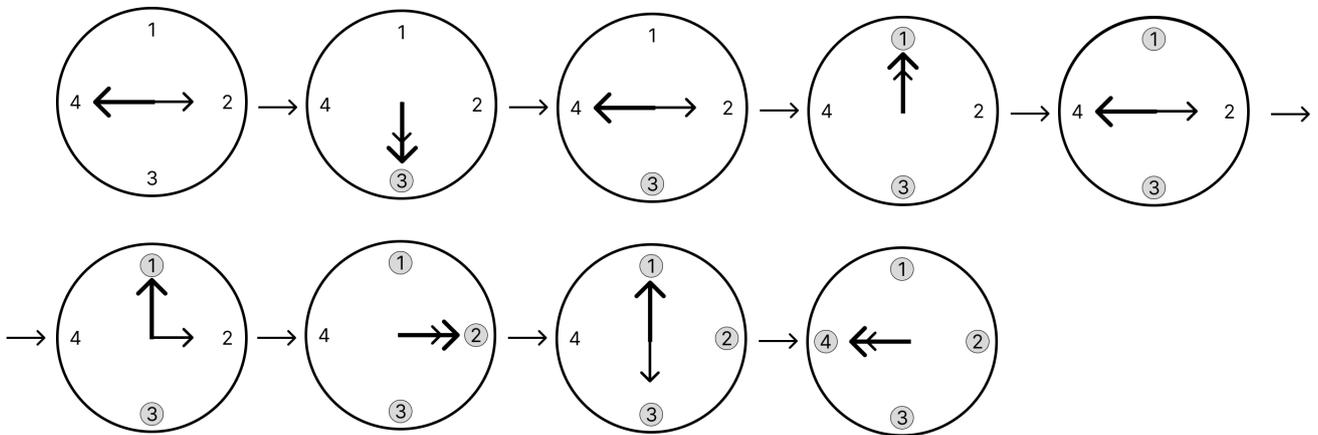
На рисунке ниже изображено состояние головоломки после каждого действия. Большая стрелка — это первая стрелка, маленькая — вторая.



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 4. 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 5. 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.

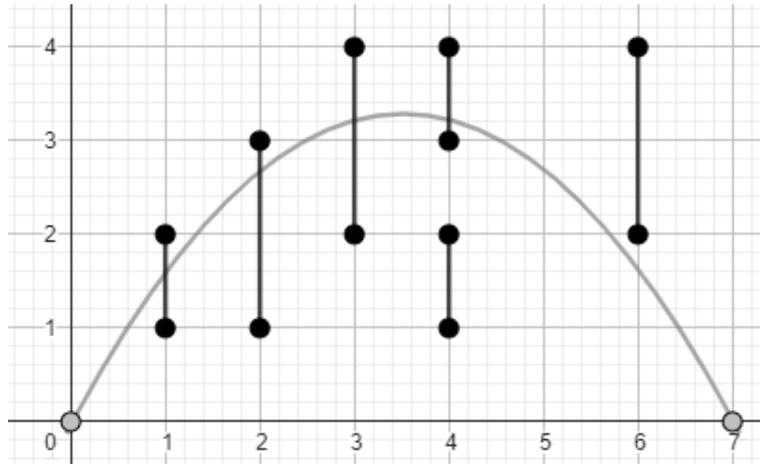
Пример

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 6. 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.