

## Задача А. Битва за Хогвартс

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Приспешники Тёмного Лорда уже подошли совсем близко к Хогвартсу.

Хогвартс окружен системой защитных башен. Профессор Флитвик накладывает защитные заклятия на замок. Заключаются они в следующем: вокруг выпуклой оболочки башен создается защитный барьер.

Тёмный Лорд, пользуясь Бузинной палочкой, может разрушить защитный барьер за минуту, при этом все башни на выпуклой оболочке тоже разрушаются.

После того, как башни уничтожены, Флитвик мгновенно восстанавливает защитный барьер на выпуклой оболочке оставшихся башен, а Вола-де-Морт их снова разрушает через минуту. Так продолжается, пока все башни не падут.

У Гарри и его друзей мало времени — они ищут и уничтожают очередной крестраж. Поэтому их очень интересует, сколько времени у них осталось.

Рассчитайте для каждой башни момент времени, когда она будет уничтожена.

### Формат входных данных

В первой строке вводится одно число  $n$  ( $1 \leq n \leq 2 \cdot 10^4$ ) — количество башен.

В следующих  $n$  строках вводятся целочисленные координаты башен —  $x_i, y_i$  ( $|x_i|, |y_i| \leq 10^4$ ).

Гарантируется, что башни расположены так, что каждый следующий защитный барьер будет лежать строго внутри предыдущего (то есть, они не пересекаются и не имеют точек касания).

### Формат выходных данных

Выведите  $n$  строк, каждая из которой содержит одно число — момент времени (в минутах), к которому падёт каждая башня, начиная с первой.

### Пример

стандартный ввод	стандартный вывод
5	1
0 0	1
4 4	1
0 4	2
1 1	1
4 0	

## Задача В. Замок

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

Есть замок — точка  $(0, 0)$ . Замок окружен несколькими непересекающимися заборами, каждый представляет из себя выпуклый многоугольник.

Есть также  $m$  захватчиков, известны их координаты. Захватчики не умеют перелезть через заборы. Захватчика будем считать опасным, если он находится внутри внешнего забора. Требуется вычислить суммарную площадь области, куда опасные захватчики могут добраться без пересечения заборов.

### Формат входных данных

В первой строке задано число  $n$  — количество заборов ( $1 \leq n \leq 100\,000$ ). Далее следуют описания  $n$  заборов. Каждое описание начинается с числа  $k$ , далее следуют  $k$  строк, содержащих по два числа  $x$  и  $y$  — координаты вершин ( $|x|, |y| \leq 2 \cdot 10^6$ ). Вершины каждого многоугольника перечисляются в порядке обхода против часовой стрелки. Гарантируется, что точка  $(0, 0)$  лежит внутри каждого забора.

Далее следует число  $m$  ( $0 \leq m \leq 100\,000$ ) — количество захватчиков. В следующих  $m$  строках заданы координаты захватчиков.

Суммарное число вершин во всех многоугольниках не превосходит  $100\,000$ .

### Формат выходных данных

Выведите единственное число — общую захваченную площадь с шестью знаками после десятичной точки.

### Пример

стандартный ввод	стандартный вывод
3 4 -10 -10 10 -10 10 10 -10 10 4 20 20 -20 20 -20 -20 20 -20 4 30 -30 30 30 -30 30 -30 -30 3 1 1 22 23 111 123	2400.000000

## Задача С. Выпуклая оболочка

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

Дано  $N$  точек на плоскости.

Нужно построить их выпуклую оболочку.

Гарантируется, что выпуклая оболочка не вырождена.

### Формат входных данных

На первой строке число  $N$  ( $3 \leq N \leq 10^5$ ). Следующие  $N$  строк содержат пары целых чисел  $x$  и  $y$  ( $-10^9 \leq x, y \leq 10^9$ ) — точки.

Будьте аккуратны! Точки произвольны. Бывают совпадающие, бывают лежащие на одной прямой в большом количестве.

### Формат выходных данных

В первой строке выведите  $N$  число вершин выпуклой оболочки. Следующие  $N$  строк должны содержать координаты вершин в порядке обхода. Никакие три подряд идущие точки не должны лежать на одной прямой. Кроме того, в последней строке выведите площадь получившейся выпуклой оболочки. Площадь необходимо вывести абсолютно точно.

### Примеры

стандартный ввод	стандартный вывод
5 0 0 2 0 0 2 1 1 2 2	4 0 0 0 2 2 2 2 0 4.0
3 0 0 0 1 1 1	3 0 0 0 1 1 1 0.5

## Задача D. Стена

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

64 мегабайта

Жил-был жадный король, который однажды приказал главному архитектору окружить королевский замок стеной. Король был настолько жаден, что не желал слушать рассказы архитектора о красивой кирпичной стене с прекрасным силуэтом и изящными высокими башнями. Вместо этого он приказал окружить замок стеной, затратив минимальное количество камня и времени, но потребовал, чтобы стена не подходила к замку ближе, чем на заданное расстояние. Если король узнает, что архитектор потратил не минимально возможное количество ресурсов, то архитектор лишится головы. Более того, король потребовал, чтобы архитектор сразу же предложил проект стены с указанием минимального количества ресурсов, необходимых для постройки.

Вы должны помочь архитектору сохранить голову, написав программу для поиска минимальной длины стены, удовлетворяющей условиям короля.

Задачу упрощает то, что замок короля имеет форму многоугольника и расположен на равнине. Архитектор уже ввел систему координат и точно измерил координаты вершин замка в футах.

### Формат входных данных

Первая строка входного файла содержит числа  $N$  и  $L$ , разделенные пробелом.  $N$  ( $3 \leq N \leq 1000$ ) — это количество вершин в королевском замке, а  $L$  ( $1 \leq L \leq 1000$ ) — минимальное количество футов, на которое стена может приближаться к замку.

Следующие  $N$  строк описывают координаты замка в порядке обхода по часовой стрелке. В каждой строке через пробел записаны целые числа  $x_i$  и  $y_i$ , разделенные пробелом ( $-10000 \leq x_i, y_i \leq 10000$ ), которые обозначают координаты  $i$ -ой вершины. Все вершины различны, и никакие две стороны не пересекаются кроме как по вершинам.

### Формат выходных данных

Выведите минимальную длину стены в футах, удовлетворяющей условиям короля с точностью не менее 6 знаков после запятой.

### Пример

стандартный ввод	стандартный вывод
9 100 200 400 300 400 300 300 400 300 400 400 500 400 500 200 350 200 200 200	1628.3185307180

## Задача Е. Теодор Рузвельт

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

«Теодор Рузвельт» — флагман военно-морского флота Кукуляндии. Заклятые враги кукуляндцев, флатландцы, решили уничтожить его. Они узнали, что «Теодор Рузвельт» представляет собой выпуклый многоугольник из  $n$  вершин и узнали его координаты. Затем они выпустили  $m$  баллистических ракет и определили координаты точек, где эти ракеты взорвались. По расчётам штаба флатландцев, «Теодор Рузвельт» будет уничтожен, если в него попадёт хотя бы  $k$  ракет. Вычислите, удалось ли флатландцам уничтожить корабль.

### Формат входных данных

В первой строке через пробел записаны целые числа  $n, m, k$  ( $3 \leq n \leq 10^5, 0 \leq k \leq m \leq 10^5$ ). В последующих  $n$  строках записаны координаты вершин многоугольника в порядке обхода против часовой стрелки. В следующих  $m$  строках записаны координаты точек. Гарантируется, что все координаты — целые числа, не превосходящие по модулю  $10^9$ .

### Формат выходных данных

Выведите «YES», если в многоугольнике или на его границе лежит по крайней мере  $k$  точек, и «NO» в противном случае.

### Пример

стандартный ввод	стандартный вывод
5 4 2 1 -1 1 2 0 4 -1 2 -1 -1 -2 -1 1 -1 0 1 2 3	YES

## Задача F. Разрезание торта

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Мама купила Пете на день рождения торт в виде выпуклого многоугольника. Торт большой и вкусный, и Петя хочет разделить его с мамой поровну. Для этого он хочет сделать один прямой разрез, причем ему будет удобнее, если этот разрез будет параллелен оси  $Oy$ . Помогите Пете определить, как ему разрезать торт.

### Формат входных данных

В первой строке записано целое число  $N$  ( $1 \leq N \leq 10000$ ) — количество вершин многоугольника. В последующих  $N$  строках записаны координаты вершин многоугольника в порядке обхода. Гарантируется, что все координаты — целые числа, не превосходящие по модулю  $10^3$ .

### Формат выходных данных

Выведите  $x$ -координату точки, через которую необходимо провести разрез, с точностью не менее  $10^{-6}$ .

### Пример

стандартный ввод	стандартный вывод
4	1.000000000
0 0	
0 2	
2 2	
2 0	

## Задача G. Сыр

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

Крешо купил вкуснейший сыр с перцем, но Степану перец не нравится, поэтому он хочет отрезать кусок, на котором не было бы перца. Сыр имеет форму выпуклого многоугольника, а каждая перчинка является точкой внутри него. Степан режет сыр только 1 раз. Он выбирает две вершины многоугольника, не являющиеся смежными, и режет по диагонали, соединяющей их. Затем Степан забирает ту из получившихся частей, на которой нет перца (ни внутри, ни на границе).

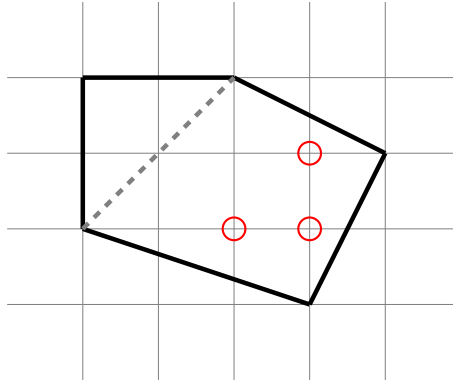


Рис. 1: Рисунок соответствует первому тесту. Пунктирной линией показан разрез Степана.

Напишите программу, которая определит, может ли Степан отрезать кусок без перца. Если он может это сделать, выведите максимальную площадь куска, который может отрезать Степан.

### Формат входных данных

Первая строка входного файла содержит одно целое число  $N$  ( $3 \leq N \leq 3 \cdot 10^5$ ) — количество вершин в многоугольнике. Каждая из следующих  $N$  строк содержит два числа  $x_i$  и  $y_i$  — координаты  $i$ -й вершины. Следующая строка содержит одно число  $M$  ( $1 \leq M \leq 3 \cdot 10^5$ ) — количество перчинок. Каждая из следующих  $M$  строк содержит два числа  $x_i$  и  $y_i$  — координаты  $i$ -й перчинки.

Вершины многоугольника заданы в порядке обхода против часовой стрелки и образуют выпуклый многоугольник. Никакие две подряд идущие стороны не параллельны.

Все перчинки расположены в различных точках и внутри многоугольника (они не расположены на стороне или снаружи многоугольника).

Все входные координаты по модулю не превосходят  $10^9$ .

### Формат выходных данных

Выведите одно число — удвоенную максимальную площадь (это число всегда целое). Если отрезать кусок без перца невозможно, выведите 0.

**Пример**

стандартный ввод	стандартный вывод
5	4
0 1	
3 0	
4 2	
2 3	
0 3	
3	
2 1	
3 1	
3 2	



## Задача Н. Фатальная ошибка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

На дворе был теплый майский вечер, Захар только что дописал свою дипломную работу, название которой настолько сложно, что может испугать неподготовленного читателя. Студент, обрадовавшись, решил распечатать написанный на компьютере текст на  $n$  прямоугольных листах бумаги размера  $h \times w$  сантиметров. После того, как дипломная работа была распечатана, Захар сложил все листы бумаги в аккуратную стопку: наверху стопки лежала первая страница текста, под ней — вторая, и так далее.

Радости Захара не было предела до тех пор, пока он не совершил фатальную ошибку, решив попить чай и, конечно же, разлив весь напиток прямоком на стопку бумаги. В результате этого на каждом листе бумаги образовалось пятно, имеющее по счастливой случайности форму выпуклого многоугольника. Пятно было одинаково хорошо заметно с каждой стороны каждого листа бумаги. Более того, так как чай медленно распространялся по стопке сверху вниз, пятно, образовавшееся на  $i$ -м сверху листе, оказалось *строго вложено* в пятно, образовавшееся на  $(i - 1)$ -м сверху листе.

Будем говорить, что многоугольник  $A$  *строго вложен* в многоугольник  $B$ , если ломаные, образующие границы многоугольников, не имеют ни одной общей точки, а также любая вершина многоугольника  $A$  находится внутри многоугольника  $B$ .

Захар был в бешенстве. Он схватил все  $n$  листов бумаги и яростно разбросал их по комнате. Немного успокоившись, студент решил собрать все листы обратно в стопку, но оказалось, что он забыл проставить номера страниц на листах, и теперь не знает, в каком порядке они шли изначально. Более того, некоторые листы во время полета по комнате неоднократно перевернулись.

Более формально, каждый лист бумаги при падении на пол мог либо оказаться в исходном состоянии, либо повернуться на 180 градусов таким образом, что текст теперь напечатан на листе не сверху вниз, а наоборот, либо перевернуться наизнанку, либо и перевернуться наизнанку, и повернуться на 180 градусов одновременно.

Захар тут же решил посчитать, сколько существует различных способов собрать стопку бумаги таким образом, чтобы пятно на  $i$ -м сверху листе оказалось строго вложено в пятно на  $(i - 1)$ -м сверху листе. Перед тем, как собрать стопку, Захар может повернуть и/или перевернуть каждый лист бумаги так, чтобы он оказался в одном из четырех описанных выше состояний.

Два способа собрать стопку считаются различными, если выполнено одно из двух условий:

1. Порядок следования листов в двух способах различается.
2. Порядок следования листов в двух способах совпадает, но есть хотя бы один лист, который находится в разных состояниях в первом и втором способе.

### Формат входных данных

Первая строка содержит три целых числа  $n$ ,  $w$  и  $h$  ( $1 \leq n \leq 10^5$ ,  $1 \leq w < h \leq 10^9$ ) — количество листов бумаги и их размеры, соответственно.

Следующие строки содержат описания пятен, образовавшихся на каждом из  $n$  листов бумаги.

Первая строка описания каждого пятна содержит целое число  $k_i$  ( $3 \leq k_i \leq 10^5$ ) — количество вершин в многоугольнике, задающем пятно, образовавшееся на  $i$ -м листе бумаги.

Каждая из следующих  $k_i$  строк содержит два целых числа  $x_{ij}$  и  $y_{ij}$  ( $0 \leq x_{ij} \leq w$ ,  $0 \leq y_{ij} \leq h$ ) — координаты вершин очередного многоугольника в порядке обхода против часовой стрелки. Будем считать, что начало координат каждого листа бумаги расположено в его левом нижнем углу.

Гарантируется, что все многоугольники являются выпуклыми, и никакие три вершины одного многоугольника не лежат на одной прямой.

Также гарантируется, что суммарное количество вершин всех многоугольников не превосходит  $10^5$ , то есть  $\sum_{i=1}^n k_i \leq 10^5$ .

## Формат выходных данных

Выведите одно целое число — количество различных способов собрать стопку бумаги таким образом, чтобы пятно на  $i$ -м сверху листе оказалось строго вложено в пятно на  $(i - 1)$ -м сверху листе.

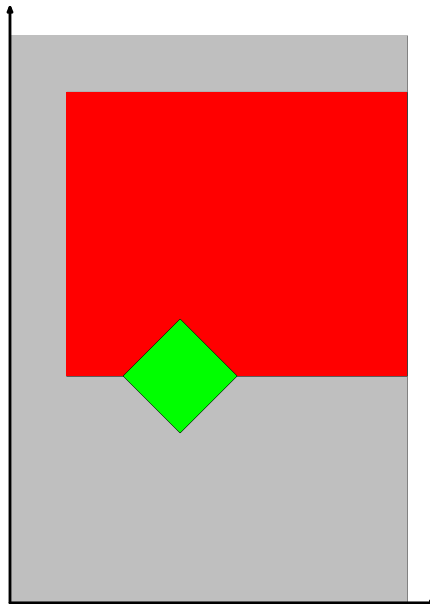
Так как ответ может быть достаточно большим, выведите остаток от деления ответа на число 998 244 353.

## Пример

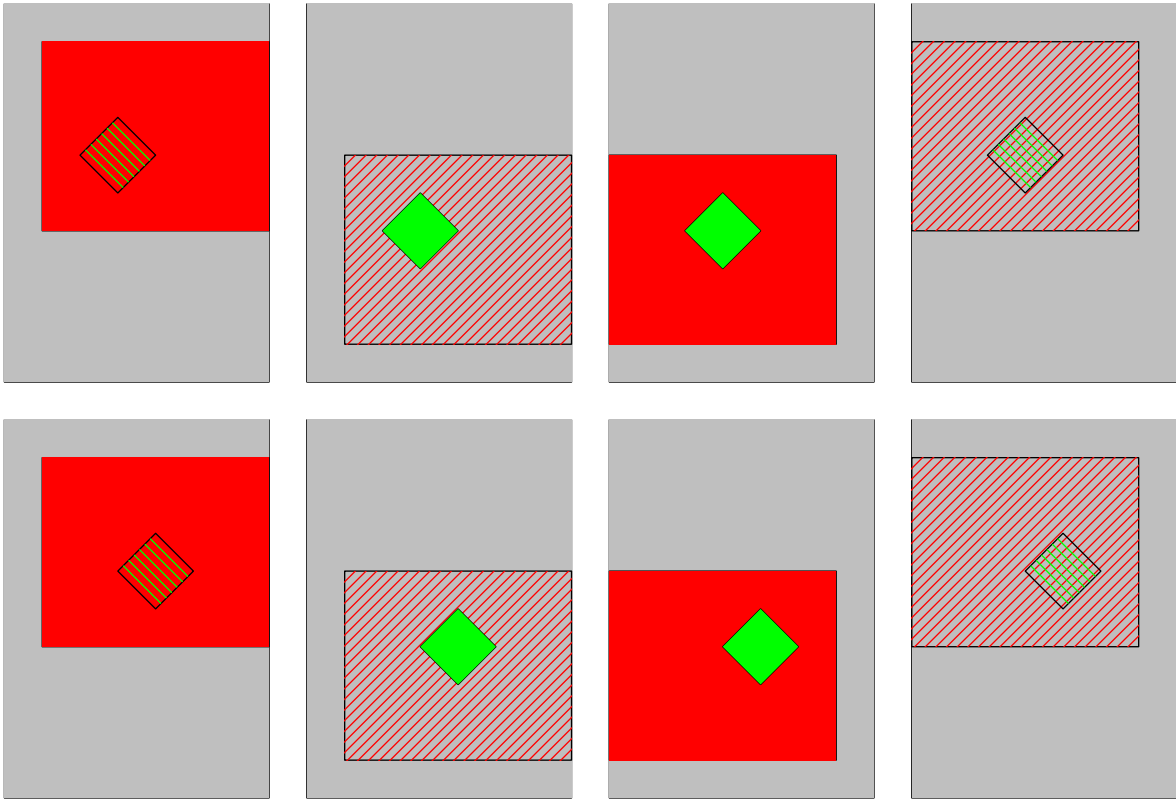
стандартный ввод	стандартный вывод
2 7 10 4 1 4 7 4 7 9 1 9 4 3 5 2 4 3 3 4 4	8

## Замечание

Рассмотрим первый пример. На рисунке ниже изображены два многоугольника в их изначальных состояниях. Для удобства восприятия многоугольники наложены друг на друга. Первый многоугольник обозначен красным цветом, а второй — зеленым цветом. Лист бумаги обозначен серым цветом.



На рисунке ниже изображены восемь способов сложить стопку из данных двух листов. Если какой-то из многоугольников заштрихован, то это означает, что лист с данным многоугольником перевернут наизнанку.



## Задача I. Разноцветные точки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Рассмотрим  $n$  точек на плоскости, пронумерованных от 1 до  $n$ , обозначим их как  $P_1, P_2, \dots, P_n$ , координаты  $i$ -й точки  $(x_i, y_i)$ .

Рассмотрим следующий процесс. Выберем номер *начальной* точки  $i$  и номер *следующей* за ней точки  $j$  ( $i \neq j$ ), а также целое число  $t$ . После этого номер *прицельной* точки  $k$  вычисляется по следующему алгоритму. Рассмотрим вектор  $\overrightarrow{P_i P_j}$ , направленный из точки  $P_i$  в точку  $P_j$ . Упорядочим все точки, кроме  $j$ -й, по углу, отсчитывая против часовой стрелки от направления вектора, равного  $\overrightarrow{P_i P_j}$ , отложенного из точки  $j$ . При равенстве угла будем упорядочивать точки по возрастанию расстояния до точки  $j$ . В качестве точки  $k$  выбирается точка, являющаяся  $t$ -й в данном порядке при нумерации с единицы. Далее точка  $j$  становится начальной, а точка  $k$  — следующей за ней, после чего, пользуясь тем же алгоритмом, вычисляется номер прицельной точки. Этот процесс повторяется до бесконечности.

Для лучшего понимания процесса рассмотрим следующий пример. Пусть имеются 6 точек, изображенных на рисунке 1, а  $t = 4$ . Пусть номер начальной точки равен 1, а номер следующей за ней точки равен 2. Отложим вектор  $\overrightarrow{P_1 P_2}$  от точки  $P_2$  и отсортируем все точки, кроме точки  $P_2$ , по углу, отсчитывая против часовой стрелки от направления данного вектора. На рисунке 2 отложенный вектор обозначен пунктирной линией, а также для удобства проведены векторы из точки  $P_2$  во все остальные точки.

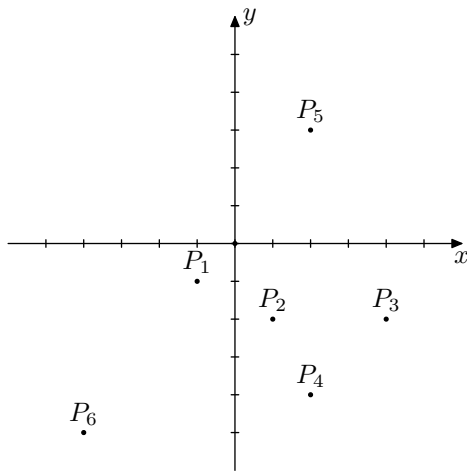


Рисунок 1: Пример множества из 6 точек

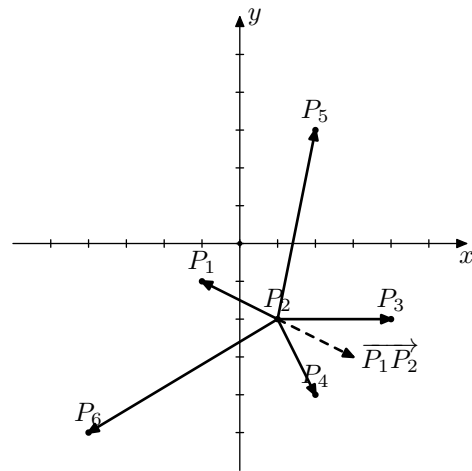


Рисунок 2: Вектор  $\overrightarrow{P_1 P_2}$ , а также векторы из точки  $P_2$  во все остальные точки

Точки будут упорядочены следующим образом:  $P_3, P_5, P_1, P_6, P_4$ . Таким образом, номер прицельной точки равен 6. Далее точка 2 становится начальной, а точка 6 — следующей.

На рисунке 3 изображен процесс для начальной точки 2 и следующей точки 6. Точки будут упорядочены следующим образом:  $P_4, P_3, P_2, P_1, P_5$ . Обратите внимание, что точка  $P_1$  в этом списке находится раньше, чем точка  $P_5$ , так как расстояние от точки  $P_1$  до точки  $P_6$  меньше, чем расстояние от точки  $P_5$  до точки  $P_6$ . Прицельная точка будет иметь номер 1.

На рисунке 4 изображен процесс для начальной точки 6 и следующей точки 1. Обратите внимание, что в данном случае вектор  $\overrightarrow{P_6 P_1}$ , отложенный из точки  $P_1$  совпадает с вектором  $\overrightarrow{P_1 P_5}$ , отложенным из точки  $P_1$ . Эти векторы изображены сплошной линией. Точки будут упорядочены следующим образом:  $P_5, P_6, P_4, P_2, P_3$ . Прицельная точка будет иметь номер 2. Таким образом, далее процесс начнется для начальной точки 1 и следующей точки 2 и зациклится.

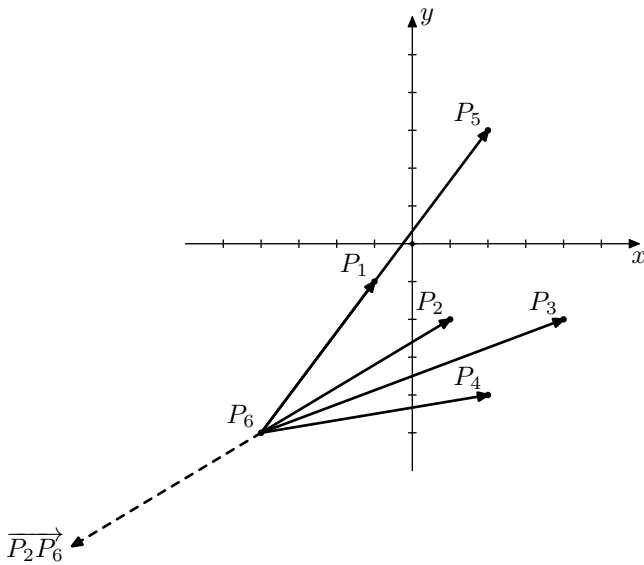


Рисунок 3: Процесс для начальной точки 2 и следующей точки 6

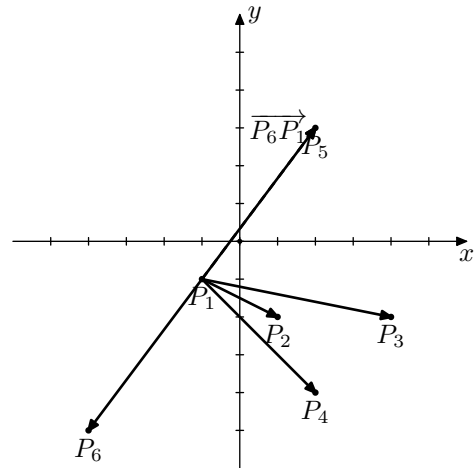


Рисунок 4: Процесс для начальной точки 6 и следующей точки 1

Покрасим каждую из  $n$  точек в один из трех цветов. Цвет  $i$ -й точки определяется следующим образом:

- Пусть существует такая точка  $j$ , что, выбрав точку  $i$  в качестве начальной, а точку  $j$  в качестве следующей, в результате описанного процесса точка  $i$  побывает начальной бесконечное количество раз. В этом случае точка  $i$  будет покрашена в **зеленый** цвет.
- Пусть точка  $i$  не была покрашена в зеленый цвет и существует такая точка  $j$ , что, выбрав точку  $i$  в качестве начальной, а точку  $j$  в качестве следующей, в результате описанного процесса точка  $i$  побывает начальной еще хотя бы один раз. В этом случае точка  $i$  будет покрашена в **синий** цвет.
- Пусть точка  $i$  не была покрашена ни в зеленый, ни в синий цвет. В этом случае точка  $i$  будет покрашена в **красный** цвет.

Для каждой точки определите, в какой цвет ее нужно покрасить.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $t$  ( $2 \leq n \leq 1000$ ,  $1 \leq t \leq n - 1$ ).

Каждая из следующих  $n$  строк содержит два целых числа  $x_i$  и  $y_i$  ( $-10^9 \leq x_i, y_i \leq 10^9$ ). Гарантируется, что никакие две точки не совпадают.

### Формат выходных данных

Выведите строку, состоящую из  $n$  символов:  $i$ -й символ строки должен обозначать цвет  $i$ -й точки. Для зеленой точки выведите букву «G», для синей точки — букву «B», а для красной точки — букву «R».

## Система оценки

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

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	10	$n \leq 10$ , все точки расположены на одной прямой		первая ошибка
2	15	все точки расположены на одной прямой	1	первая ошибка
3	10	$n \leq 10$ , гарантируется, что нет синих точек		первая ошибка
4	10	$n \leq 10$	1, 3	первая ошибка
5	15	$n \leq 100$ , гарантируется, что нет синих точек	3	первая ошибка
6	15	$n \leq 100$	1, 3, 4, 5	первая ошибка
7	5	$n \geq 3$ , все точки являются вершинами строго выпуклого многоугольника и даны в порядке обхода против часовой стрелки		первая ошибка
8	20	нет	1–7	первая ошибка

## Примеры

стандартный ввод	стандартный вывод
6 4 -1 -1 1 -2 4 -2 2 -4 2 3 -4 -5	GGBBRG
2 1 1 1 2 2	GG

## Пояснение к примеру

Рассмотрим некоторые точки из первого примера.

Точка  $P_1$  окрашена в зеленый цвет, потому что можно выбрать точку  $P_2$  в качестве следующей, и процесс посетит точку  $P_1$  бесконечное количество раз. Данный пример был рассмотрен выше в условии задачи.

Можно показать, что точка  $P_3$  не является зеленой, однако она является синей, так как можно выбрать точку 1 в качестве следующей, точка 3 окажется начальной еще хотя бы один раз. Процесс для начальной точки 1 и следующей точки 3 проиллюстрирован на рисунках 5, 6 и 7 ниже.

Для начальной точки 3 и следующей точки 1 точки будут упорядочены следующим образом:  $P_6, P_4, P_2, P_3, P_5$ . Точка с номером 3 становится прицельной. Далее для начальной точки 1 и следующей точки 3 точки будут упорядочены следующим образом:  $P_5, P_1, P_2, P_6, P_4$ . Точка с номером 6 становится прицельной. Наконец, для начальной точки 3 и следующей точки 6 точки будут упорядочены следующим образом:  $P_4, P_3, P_2, P_1, P_5$ . Точка с номером 1 становится прицельной. Далее

процесс продолжится с начальной точкой 6 и следующей точкой 1. Из примера, описанного выше в условии задачи, мы знаем, что процесс заикнется, посещая точки с номерами 6, 1 и 2.

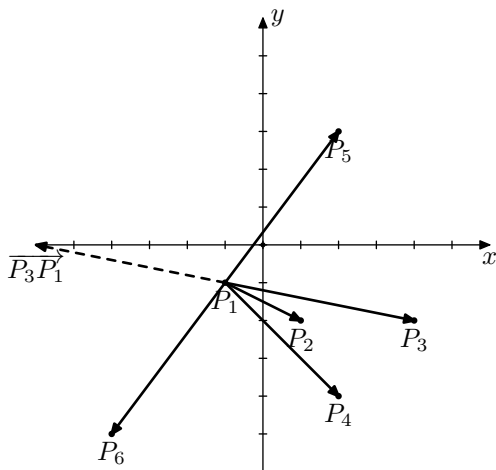


Рисунок 5: Процесс для начальной точки 3 и следующей точки 1

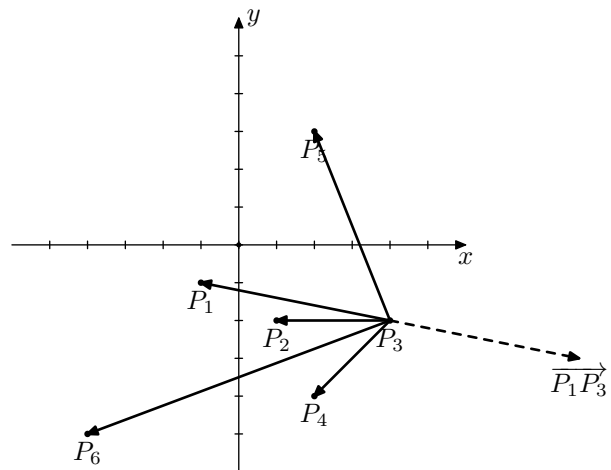


Рисунок 6: Процесс для начальной точки 1 и следующей точки 3

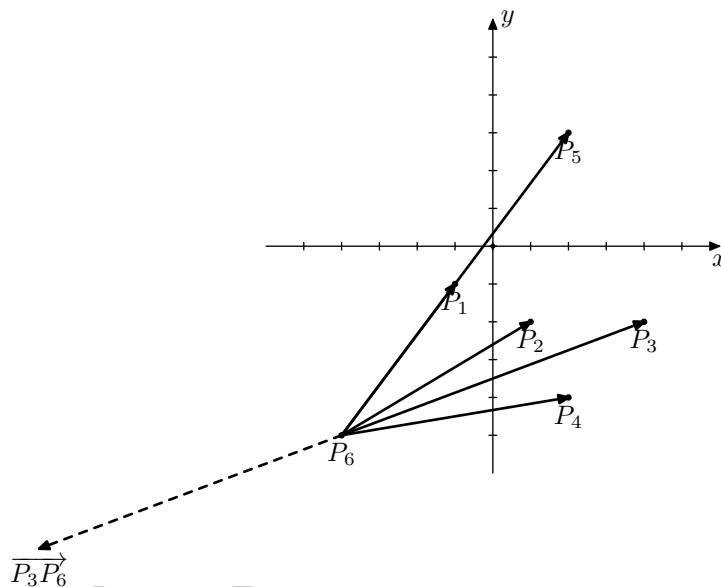


Рисунок 7: Процесс для начальной точки 3 и следующей точки 6

Во втором примере из условия легко показать, что если одна из точек является начальной, а другая — следующей, то прицельной станет точка, которая являлась начальной. Поэтому обе точки будут окрашены в зеленый цвет.

## Задача J. Львы (но не тигры!)

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

Иван — профессиональный фотограф. В погоне за хорошим кадром он готов ехать в любую точку мира. И сейчас он приехал в Уганду с целью сфотографировать львов в их естественной среде обитания. К сожалению, львов не очень заинтересовали профессиональные качества Ивана, зато их весьма заинтересовали его вкусовые качества.

Для простоты будем считать саванну бесконечной плоскостью, а Ивана и львов — точками. В начальный момент времени Иван выбирает направление, в котором он будет ехать, спасаясь от львов, и дальше все время едет по прямой. Львы, будучи великолепными охотниками, сразу определяют направление, выбранное Иваном, и планируют его поимку соответственно. В частности, львы понимают, что Иван поедет по прямой. Скорости всех львов постоянны и равны между собой. Как мы уже увидели, Иван не очень умный, поэтому сфотографировать львов он отправился в медленном автомобиле, скорость которого также постоянна и равна скорости львов.

Помогите Ивану выбрать такое направление, чтобы его не догнал ни один лев.

### Формат входных данных

В первой строке входных данных дано целое число  $n$  — количество львов ( $1 \leq n \leq 5 \cdot 10^5$ ).

Во второй строке даны два целых числа — координаты Ивана.

Далее следуют  $n$  строк, в каждой из которых даны два целых числа — координаты львов.

Все координаты не превосходят  $10^8$  по абсолютной величине. Гарантируется, что все точки во входных данных различны.

### Формат выходных данных

Если Иван не может убежать, выведите одно слово «NO» (без кавычек). Иначе в первой строке выведите одно слово «YES» (тоже без кавычек).

Составителю контеста было лень написать чекер, поэтому восстанавливать направление не требуется.

### Система оценки

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

Подзадача	Баллы	Дополнительные ограничения	Оценка	Необходимые подзадачи
0	0	Тесты из условия	подзадача	—
1	37	$1 \leq n \leq 5000$	подзадача	—
2	63	—	подзадача	1

### Примеры

стандартный ввод	стандартный вывод
2 0 0 1 1 2 2	YES
4 0 0 1 1 -1 -1 1 -1 -1 1	NO



## Задача К. Ахтунг!

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Ахтунг! Механическую няню кто-то включил, и теперь она бежит за Пином, пытаясь окружить его лаской и заботой! Во дворе Пина есть  $n$  бункеров, и он рад бы спрятаться в одном из них, но в каждом бункере у него есть очень важное и очень срочное дело.

Поэтому Пин просит вас составить маршрут его передвижения между бункерами, чтобы он смог посетить каждый ровно один раз и вернуться в начало. При этом начать Пин может с любого бункера.

Кроме того, так как Пин сам писал программу перехвата для няни и собирал её двигатель, то он точно знает, что будет пойман, если будет бежать от одного бункера к другому не по прямой, либо если он дважды пробежит через одно и то же место, то есть пересечёт или коснётся отрезка пути, который он уже пробежал.

### Формат входных данных

Во входном файле дано описание двора Пина.

В первой строке входного файла находится одно целое число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество бункеров во дворе Пина.

В следующих  $n$  строках записано по два целых числа  $x_i$  и  $y_i$  ( $|x_i|, |y_i| \leq 10^9$ ) — координаты входа в  $i$ -й бункер. Вход в бункер настолько мал по сравнению с размерами двора, что считается точкой. Никакие два бункера не лежат в одной точке.

### Формат выходных данных

В выходной файл выведите перестановку из  $n$  чисел — номера бункеров в порядке их посещения Пином, либо «No solution», если не существует маршрута, по которому Пин может пробежать и не быть пойманным няней.

### Примеры

<code>stdin</code>	<code>stdout</code>
4 0 0 0 1 1 0 1 1	3 4 2 1
3 0 0 0 1 0 2	No solution
1 0 0	1

## Задача L. Принцесса

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

Принцесса Евлампия живет в замке, окруженном забором. Жизнь принцессы тяжела, но при этом и очень интересна. Главным ее развлечением является общение с многочисленными поклонниками, постоянно прибывающими из соседних замков, городов и даже королевств.

Замок принцессы окружен забором, представляющим из себя выпуклый многоугольник. Отец принцессы, король, достаточно строг, поэтому всем поклонникам принцессы приходится попадать туда через единственную во всем заборе дырку, вместо того, чтобы войти на территорию замка через парадные ворота. Дырка находится в одной из вершин многоугольника. При этом, если пройти напрямую к дырке поклоннику не удастся, ему придется обходить забор вдоль его периметра. Естественно, каждому поклоннику интересно, сколько ему придется пройти, чтобы попасть из точки своего начального местоположения к дырке, и все спрашивают об этом принцессу, перед тем как прийти к ней в гости.

Принцесса составила список начальных местоположений всех своих поклонников и описание забора вокруг замка. Вам необходимо для каждого поклонника сообщить длину кратчайшего пути от точки его начального положения до точки, в которой находится дырка. При этом, естественно, ни одна точка этого пути не должна лежать внутри многоугольника, представляющего забор, но может лежать на его границе.

### Формат входных данных

В первой строке входного файла находятся два целых числа  $n$  и  $k$  ( $3 \leq n \leq 100\,000$ ,  $1 \leq k \leq n$ ) — количество вершин в многоугольнике, представляющем забор, и номер вершины, в которой находится дырка. В следующих  $n$  строках содержатся пары целых чисел  $x_i$  и  $y_i$ , описывающих координаты вершин многоугольника в порядке обхода против часовой стрелки.

В следующей строке дано одно целое число  $m$  ( $1 \leq m \leq 100\,000$ ) — количество поклонников принцессы. В следующих  $m$  строках содержатся пары целых чисел  $x_i$  и  $y_i$ , описывающих координаты начального положения очередного поклонника.

Все координаты не превышают  $10^9$  по абсолютной величине.

### Формат выходных данных

Для каждого поклонника выведите одно число — ответ на задачу. Ответ должен отличаться от правильного не более, чем на  $10^{-5}$ .

### Примеры

стандартный ввод	стандартный вывод
4 2	3.23606797749979
0 1	2.0
0 0	
1 0	
1 1	
2	
2 2	
-2 0	

## Задача М. Точка в многоугольнике

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

### Формат входных данных

В первой строке содержится три числа —  $N$  ( $3 \leq N \leq 100\,000$ ) и координаты точки. Последующие  $N$  строк содержат координаты углов многоугольника. Координаты — целые, не превосходят  $10^6$  по модулю.

### Формат выходных данных

Одна строка `YES`, если заданная точка содержится в приведённом многоугольнике или на его границе, и `NO` в противном случае.

### Примеры

<code>stdin</code>	<code>stdout</code>
3 2 3 1 1 10 2 2 8	YES