

Задача А. Вампирский баскетбол

Имя входного файла: `basket.in`

Имя выходного файла: `basket.out`

Как известно, вампиры любят и хорошо умеют играть в разные спортивные игры. Но так как сейчас осень, и играть на улице в бейсбол уже прохладно, вампиры решили собраться в спортивном зале и поиграть в баскетбол. Как только они попали в зал, то сразу поделились на две команды. Эдвард всегда играет на позиции разыгрывающего защитника, потому что любит давать пасы. Для определенной расстановки игроков на поле он хочет знать, кто из его команды находится в наиболее выгодной позиции, а кто нет.

Поле представляет из себя полуплоскость, каждая точка которой задается целыми координатами x и y такими, что координата y неотрицательна. Кольцо находится в точке $(0, 0)$. Также есть трехочковая дуга, представляющая полуокружность с радиусом r и центром в точке, в которой находится кольцо. Если расстояние игрока до кольца строго больше, чем r , то его забитый бросок считается за три очка, иначе за два. Эдвард знает, что все игроки его команды бросают одинаково. Каждый игрок команды противника обладает защитными навыками, измеряемыми натуральными числами.

Каждый вампир имеет майку, на которой написан его номер, причем все номера для игроков одной команды различны и идут подряд. Таким образом, если в команде Эдварда три вампира, а в команде противников два, то они будут иметь номера 1, 2, 3 и 1, 2. Сам Эдвард носит майку с номером 0, и собственная позиция ему безразлична. Эдвард считает выгодность позиции вампира из его команды под номером i по формуле

$$b_i = \frac{p_i}{d_i^2 \cdot (1 + \sum z_j)}$$

, где p_i — количество очков, которые можно получить, забив с этой точки, d_i — расстояние от вампира до кольца, а $\sum z_j$ — сумма мер защитных навыков противников, расстояние от которых до нападающего не превышает l .

Чтобы помочь Эдварду, нужно найти последовательность номеров вампиров из его команды в порядке убывания выгодности их позиции. При равенстве выгодностей позиций вампиры идут по возрастанию номера. Тогда Эдвард наконец-то сможет решить, кому же он хочет отдать пас.

Формат входного файла

В самой первой строке заданы числа n, m, r, l ($1 \leq n, m \leq 10^3$, $1 \leq r, l \leq 10^5$) — количество вампиров в команде Эдварда, количество противников, радиус трехочковой дуги, расстояние, определяющее влияние защитников на игроков.

В следующих n строках перечислены координаты сокомандников Эдварда x_i и y_i ($1 \leq y_i \leq 10^5$, $-10^5 \leq x_i \leq 10^5$) в порядке возрастания номеров на майках. В следующих m строках перечислены координаты противников x_j, y_j ($1 \leq y_j \leq 10^5$, $-10^5 \leq x_j \leq 10^5$) также в порядке возрастания номеров и z_j ($1 \leq z_j \leq 100$) — мера защитного умения противника под номером j .

Все позиции всех игроков различны.

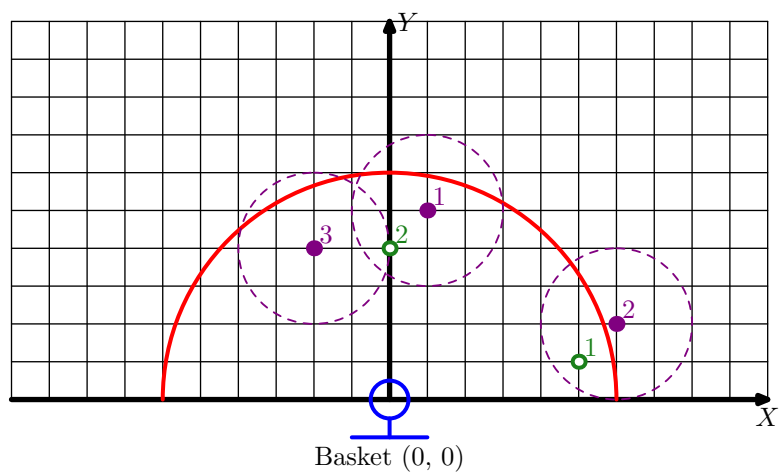
Формат выходного файла

Выведите в выходной файл n чисел — последовательность номеров вампиров в нужном порядке.

Пример

<code>basket.in</code>	<code>basket.out</code>
3 2 6 2 1 5 6 2 -2 4 5 1 2 0 4 3	2 3 1

Примечание



На картинке изображен пример. $b_1 = \frac{2}{26 \cdot 4} = \frac{1}{52}$, $b_2 = \frac{3}{40 \cdot 3} = \frac{1}{40}$, $b_3 = \frac{2}{20 \cdot 4} = \frac{1}{40}$.
Обратите внимание, что кольцо является точкой.

Задача В. Место преступления

Имя входного файла: `crimescene.in`
 Имя выходного файла: `crimescene.out`

Хотя инспектор Лестрейд не отличается выдающимися дедуктивными способностями, он в совершенстве овладел всеми деталями рутинной полицейской работы. Как ни странно, многие стандартные процедуры могут нести в себе загадку, достойную не меньшего внимания, чем само преступление.

Однажды вечером инспектор прибыл на место преступления — огороженный забором пустырь. Инспектор заметил, что пустырь представлял собой выпуклый многоугольник, в вершинах которого стояли столбы. По всей видимости, раньше между каждыми двумя столбами, стоящими в смежных вершинах многоугольника, существовала секция забора, ограждавшего пустырь. Однако, время неумолимо, и некоторые секции забора исчезли в неизвестных направлениях. Инспектор заметил, что у любой исчезнувшей секции забора обязательно присутствуют секции, смежные с ней.

Затем Лестрейд приступил к ограждению места преступления полицейской лентой. Тут он обнаружил крайне неприятное обстоятельство: у него осталось только l метров ленты. Инспектор считает, что ленту не стоит резать на несколько кусков, и что она должна находиться только по периметру пустыря. Это означает, что инспектор зафиксирует в какой-то точке на границе пустыря один из концов ленты, после чего пройдет вдоль границы пустыря l метров в одном из двух возможных направлений, после чего зафиксирует второй конец ленты в той точке, где окажется в тот момент. Понятно, что вся пройденная им часть границы пустыря окажется закрыта лентой, а вся остальная часть — нет.

Теперь Лестрейд хочет знать, какую минимальную длину границы пустыря, не закрытую секциями забора, ему не удастся закрыть и лентой. Помогите ему, ведь до прибытия на место преступления Шерлока Холмса осталось совсем немного времени.

Формат входного файла

В первой строке входного файла содержится три целых числа n , l и k ($3 \leq n \leq 10^5$, $0 \leq l \leq 10^{18}$, $0 \leq k \leq \frac{n}{2}$) — количество вершин многоугольника, который представляет собой пустырь, длина ленты инспектора и количество дыр в заборе.

В следующей строке находится k целых чисел a_i ($1 \leq a_i \leq n$, $a_i < a_{i+1}$), описывающие отсутствующие стороны многоугольника. Каждому a_i соответствует отсутствие стороны между вершинами с номерами a_i и $a_i \bmod n + 1$. Гарантируется, что нет двух подряд идущих отсутствующих сторон.

В следующих n строках находится по два целых числа x_i и y_i ($|x_i| \leq 10^{18}$, $|y_i| \leq 10^{18}$) — координаты i -й вершины. Вершины заданы в порядке одного из двух возможных обходов многоугольника.

Формат выходного файла

В выходной файл выведите одно вещественное число — минимальную суммарную длину дыр в заборе, которые не удастся закрыть лентой. Ответ будет считаться правильным, если он отличается от правильного не более, чем на $p \times 10^{-6}$, где p — периметр многоугольника.

Примеры

<code>crimescene.in</code>	<code>crimescene.out</code>
<pre> 6 4 3 1 3 5 0 0 3 0 4 1 3 2 0 2 -1 1 </pre>	<pre> 2.8284271 </pre>

Задача С. Наследство

Имя входного файла: `inheritance.in`
Имя выходного файла: `inheritance.out`

Среди городов Средиземноморья особенно известен город Сомелот. В городе регулярно проводятся пиры, рыцарские турниры и балы. А для защиты от непрошенных гостей в городе было построено n смотровых башен, с которых зоркие воины следят за окрестностями. Долгое время городом правил мудрый лорд Эревард. Во времена его правления город процветал, а жители города любили и уважали своего правителя. Но время не пощадило и его, и на девяносто третьем году жизни он скончался.

По завещанию, управление городом перешло к его сыновьям: Бенедикту и Теобальту. Но политические взгляды братьев не совпали, они крепко разругались и решили разделить город. Братья решили построить два забора, которые будут отделять новые города друг от друга. Заборы должны состоять из секций, каждая из которых соединяет две смотровых башни. Также заборы должны быть замкнутыми, и каждая башня должна стоять на конце ровно двух секций забора. Ни одна башня не должна остаться вне или внутри новых городов.

Кроме этого, чтобы как можно реже встречаться друг с другом, братья хотят, чтобы у заборов не было общих точек как на границах, так и внутри самих заборов. А также, чтобы в городе с каждой смотровой башни было видно все точки, находящиеся внутри соответствующего города. Считается, что точку видно с башни, если на отрезке между башней и этой точкой нет заборов и других башен.

Для выполнения этой работы были наняты лучшие архитекторы, но они не смогли решить одну простую задачу: как разделить город так, чтобы никому из братьев не было обидно. Архитекторы обратились с этим вопросом к вам. Вам необходимо построить забор так, чтобы разница площадей новых городов была минимальна.

Формат входного файла

Первая строка входного файла содержит число n ($6 \leq n \leq 100$) — число смотровых башен в городе Сомелот. Следующие n строк содержат по два числа x_i, y_i ($1 \leq x_i, y_i \leq 10^9$) — координаты i -ой смотровой башни. Гарантируется, что никакие три башни не лежат на одной прямой и никакие две башни не лежат в одной точке.

Формат выходного файла

В первой и единственной строке выходного файла выведите минимально возможную разность площадей новых городов, с точностью не менее 6 знаков после запятой. Если разделить город так, чтобы были выполнены требования братьев невозможно, выведите -1.

Примеры

<code>inheritance.in</code>	<code>inheritance.out</code>
6 0 0 0 1 1 0 3 2 2 2 3 1	0

Задача D. Чертеж

Имя входного файла: `painting.in`
 Имя выходного файла: `painting.out`

Убегая от очередного недружелюбного существа из параллельного измерения, Гордон Фримен укрылся в лаборатории, где раньше изучали поведение различных кристаллов, доставленных из измерения Зен. Когда он понял, что ему не угрожает никакая опасность, Фримен осмотрелся и увидел интересное оборудование, которое предназначалось для сверхточного рисования сложных схем. Однако, оно не было рассчитано на рисования обычных чертежей, и не могло нарисовать некоторые из них в силу своего устройства. На компьютере, который был в лаборатории, Фримен нашел чертежи всех помещений федерального исследовательского центра, наличие которых в распечатанном виде очень сильно увеличило бы его шансы на выживание. Нетрудно догадаться, что он решил сделать!

Само устройство по рисованию схем выглядело следующим образом. В некоторой точке A_0 плоскости чертежа была закреплена «рука», состоящая из n последовательно соединенных с помощью шарниров металлических сегментов. Обозначим длину первого сегмента (того, который прикреплен к точке A_0) как l_1 , следующего l_2 , ..., последнего l_n . Обозначим точки соединения сегментов с номерами i и $i + 1$ как A_i , а конец последнего сегмента A_n . В точке A_n закреплён инструмент, который позволяет рисовать точку на чертеже. Устройство управления может разместить шарнирные соединения в точках A_i в плоскости чертежа, если для всех $i < n$ справедливо, что $|A_i A_{i+1}| = l_i$.

Чтобы нарисовать чертежи, Фримену необходимо узнать, все ли необходимые точки сможет нарисовать это устройство. Также для каждой точки ему необходимо узнать координаты всех A_i . Помогите ему в этом.

Формат входного файла

В первой строке входного файла находятся четыре целых числа x_0, y_0, x_n, y_n ($-10\,000 \leq x_0, y_0, x_n, y_n \leq 10\,000$) — координаты точки A_0 , к которой прикреплен первый сегмент чертежного устройства, а также точки A_n , которую необходимо нарисовать.

В следующей строке дано целое число n ($1 \leq n \leq 10^3$) — количество сегментов в устройстве рисования. В следующей строке находятся n целых чисел l_i ($1 \leq l_i \leq 1\,000$) — длины сегментов.

Формат выходного файла

В первой строке выходного файла выведите YES, если можно нарисовать точку A_n и NO в противном случае. Если точку нарисовать можно, то в следующих $n + 1$ строках выведите координаты A_i для всех i от нуля до n . Ответ будет засчитан, если расстояние между выведенными вами точками A_i и A_{i+1} отличается от l_i не более, чем на 10^{-3} .

Примеры

painting.in	painting.out
0 0 6 0	YES
2	0.000 0.000
5 5	3.000 4.000
	6.000 0.000

Примечание

Количество тестов, в которых $\sum_1^n l_i$ не превышает расстояния от точки A_0 до точки A_n , составляет 20 процентов от общего числа тестов.

Количество тестов, в которых $n \leq 2$, составляет 40 процентов от общего числа тестов.