

01. Выпуклая оболочка точек плоскости

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

Задано множество точек плоскости. Требуется найти минимальный по включению выпуклый многоугольник, содержащий все точки из заданного множества.

Формат входного файла

На первой строчке входного файла записано число n — количество точек в множестве. На последующих n строчках записаны пары вещественных чисел (x_i, y_i) , задающие координаты элементов множества на плоскости.

Формат выходного файла

На первой строке выходного файла выведите m — количество вершин многоугольника. Далее, на m строчках выведите координаты точек, являющихся вершинами многоугольника в порядке обхода против часовой стрелки.

Примеры

standard input	standard output
5	4
0.0 0.0	0.0 0.0
0.0 1.0	1.0 0.0
1.0 0.0	1.0 1.0
1.0 1.0	0.0 1.0
0.5 0.5	

02. Растеризация отрезка

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

Пусть на \mathbb{R}^2 задана координатная сетка: плоскость разбита на непересекающиеся ячейки вида $[n; n+1) \times [m; m+1)$, где n и m — целые числа. Растеризацией объекта назовём последовательность ячеек, пересекающих отрезок, упорядоченную лексикографически по координатам (x, y) нижнего левого угла.

Формат входного файла

В первых двух строчках входных данных заданы концы отрезка своими координатами.

Формат выходного файла

В первой строчке выходных данных требуется вывести количество ячеек плоскости, которые пересекает исходный отрезок. Далее требуется вывести координаты нижнего левого угла для каждой ячейки плоскости, входящей в растеризацию отрезка, на новой строчке для каждой ячейки.

Примеры

standard input	standard output
0.5 0.5 2.5 2.5	3 0 0 1 1 2 2
0.5 0.5 2.5 -1.5	5 0 0 1 0 1 -1 2 -1 2 -2

03. Поиск множества точек плоскости, попавших в прямоугольник

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

На плоскости задано множество из n точек. Также задано m прямоугольников, стороны которых параллельны осям координат. Для каждого прямоугольника требуется вывести все точки множества, которые он содержит.

Формат входного файла

В первой строчке входного файла записано натуральное число n — количество точек в множестве. На последующих n строчках через пробел записаны пары вещественных чисел (x_i, y_i) , задающих координаты точек на плоскости. На следующей строчке записано натуральное число m — количество прямоугольников. На последующих m строчках записаны по две пары вещественных чисел (a_i, a_y, b_x, b_y) , задающих координаты левого нижнего угла и правого верхнего угла i -го прямоугольника соответственно.

Формат выходного файла

В выходной файл выведите m блоков. В первой строчке каждого блока выведите число c_i — количество точек, попавших в i -ый прямоугольник. В следующие c_i строчек выведите координаты этих точек.

Примеры

standard input	standard output
8	8
1 -1	1 -1
2 3	2 3
4 5	4 5
4 2	4 2
-1 2	-1 2
-2 3	-2 3
0 3	0 3
-2 -1	-2 -1
4	0
-3 - 3 5 6	3
2 0 3 1	2 3
2 2 4 5	4 5
-2 -1 0 3	4 2
	4
	-1 2
	-2 3
	0 3
	-2 -1

04. Принадлежность точки произвольному многоугольнику

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

На плоскости заданы точки и некоторый произвольный многоугольник с дырками. Необходимо определить, принадлежат ли они этому многоугольнику.

Формат входного файла

Первая строка входного файла содержит целое число n — количество вершин многоугольника. В последующих n строчках содержатся пары вещественных чисел (x_i, y_i) — координаты i -ой вершины многоугольника. В следующей строке содержится целое число m — количество дырок в многоугольнике. В последующих m строчках содержится информация о дырках. В j -ой строке содержится число m_j — количество вершин в дырке, а затем m_j пар вещественных чисел (x_{m_j}, y_{m_j}) — координаты этих вершин. Следующая строка содержит целое число l — количество точек, для которых требуется ответить, принадлежат ли они многоугольнику. В последующих l строчках содержатся пары вещественных чисел (x_k, y_k) — координаты точек. Вершины многоугольника заданы в порядке обхода против часовой стрелки, вершины дырок — по часовой стрелке.

Формат выходного файла

В выходной файл выведите l строк, содержащих 1, если k -ая точка внутри многоугольника, 0 — на границе, -1 — снаружи.

Примеры

standard input	standard output
4 0 0 10 0 10 10 0 10 0 1 10 4	0
3 0 0 10 0.5 0 1 1 3 0.1 0.1 0.1 0.9 9 0.5 1 5 0.5	-1

05. Массовая принадлежность точки выпуклому прямоугольнику

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

На плоскости задан планарный граф. Требуется на каждый запрос локализации выдать ячейку минимальной размерности, содержащую точку запроса.

Формат входного файла

В первой строчке входного файла записано три натуральных числа n_p , n_s и n_f — количество вершин, отрезков и граней графа соответственно. На последующих n_s строчках заданы вершины. Вершины задаются парами вещественных чисел (x_i, y_i) , задающих координаты точек. На следующих n_s строчках заданы отрезки. Каждый отрезок задается номерами двух вершин. На следующих n_f строчках задаются грани. Каждая грань задается количеством отрезков образующих её и номерами этих отрезков. На следующей строчке записано натуральное число m — количество запросов. На последующих m строчках следуют пары вещественных чисел (x_i, y_i) , задающих координаты точек-запросов. Объекты нумеруются с 0 в порядке появления во входном файле.

Формат выходного файла

В выходной файл выведите два натуральных числа: размерность ячейки и ее номер. Если точка лежит во внешней области вывести -1.

Примеры

standard input	standard output
6 11 6	1 4
10 10	2 1
10 -10	0 3
-10 10	-1
-10 -10	
5 - 5	
-5 5	
0 1	
1 3	
3 4	
0 4	
4 5	
3 5	
5 0	
2 5	
2 3	
2 0	
4 1	
3 7 9 6	
3 3 6 4	
3 7 5 8	
3 2 4 5	
3 1 2 10	
3 0 3 10	
4	
0 0	
5 5	
-10 -10	
100 100	

06. Растеризация треугольника

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

Пусть на \mathbb{R}^2 задана координатная сетка: плоскость разбита на непересекающиеся ячейки вида $[n; n+1) \times [m; m+1)$, где n и m — целые числа. Растеризацией объекта назовём множество ячеек, перечисленных в лексикографическом порядке, которые лежат внутри или на границе объекта. В данной задаче требуется найти растеризацию треугольника.

Формат входного файла

На трех строчках через пробел записаны пары вещественных чисел (x_i, y_i) , задающих координаты вершин треугольника на плоскости. Вершины обходятся против часовой стрелки.

Формат выходного файла

В первой строчке выходных данных требуется вывести количество ячеек плоскости, которые участвуют в растеризации. Далее требуется на каждой строчке вывести ячейки плоскости (ячейка задаётся координатами нижнего левого угла) в лексикографическом порядке.

Примеры

standard input	standard output
0.5 0.5	9
4 0	0 0
3 3	1 0
	1 1
	2 0
	2 1
	2 2
	3 0
	3 1
	3 2

07. Растеризация выпуклого многоугольника

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

Пусть на \mathbb{R}^2 задана координатная сетка: плоскость разбита на непересекающиеся ячейки вида $[n; n+1) \times [m; m+1)$, где n и m — целые числа. Растеризацией объекта назовём множество ячеек, перечисленных в лексикографическом порядке, которые лежат внутри или на границе объекта. В данной задаче требуется найти растеризацию выпуклого многоугольника.

Формат входного файла

На первой строчке входного файла записано натуральное число n — количество вершин многоугольника. На последующих n строчках через пробел записаны пары вещественных чисел (x_i, y_i) , задающих координаты вершин на плоскости. Вершины обходятся против часовой стрелки.

Формат выходного файла

В первой строчке выходных данных требуется вывести количество ячеек плоскости, которые участвуют в растеризации. Далее требуется на каждой строчке вывести ячейки плоскости (ячейка задаётся координатами нижнего левого угла) в лексикографическом порядке.

Примеры

standard input	standard output
4	12
0.5 0.5	0 0
4.5 0.5	1 0
4.5 2.5	1 1
2.5 2.5	2 0
	2 1
	2 2
	3 0
	3 1
	3 2
	4 0
	4 1
	4 2

08. Поиск множества точек d -мерного пространства, попавших в ячейку пространства

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

Дана ячейка и точки пространства \mathbb{R}^d . Необходимо определить, какие из них лежат внутри данной ячейки.

Формат входного файла

Первая строка входного файла содержит натуральное число d — размерность пространства. В следующих двух строках содержатся $2*d$ вещественных чисел, по d в каждой, — точки, описывающие ячейку. Во второй строке — левая нижняя точка, в третьей — правая верхняя. Четвертая строка содержит натуральное число n — количество точек, для которых нужно определить, попали ли они в ячейку. Последующие n строк содержат по d вещественных чисел — координаты точек.

Формат выходного файла

В первую строку выходного файла выведите целое неотрицательное число m — количество точек, попавших в ячейку. В следующей строке укажите номера этих точек. Точки нумеруются с единицы в порядке появления во входных данных.

Примеры

standard input	standard output
2 0 0 2 2 6 0 8 1 6 1.5 1.5 0.5 0.5 10 10 0 0	3 3 4 6
3 0 0 0 3 3 3 5 1 1 1 68 37.5 100 0 0 0 1.5 0.5 2 3.5 3.1 3.2	3 1 3 4

09. Упрощение полигональной цепи

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

Полигональная цепь — упорядоченное множество точек, последовательно соединенных отрезками. Ваша задача — упростить полигональную цепь с помощью алгоритма Дугласа-Пеккера (*Douglas-Peucker*) так, чтобы расстояние от исходной цепи до получившейся было не больше ε .

Формат входного файла

На первой строке входного файла n — количество вершин полигональной цепи. На последующих n строчках записаны пары вещественных чисел (x_i, y_i) , задающие координаты вершин. В последней строке записано число ε .

Формат выходного файла

На первой строке выходного файла выведите m — количество вершин в упрощенной полигональной цепи. На второй строке через пробел выведите m чисел — номера вершин, используемых в упрощенной полигональной цепи, в возрастающем порядке. Нумерация точек начинается с единицы и задаётся в порядке следования их координат во входном файле.

Примеры

standard input	standard output
5 0 0 3 2 5 4 2 5 0 8 1.4142135	3 1 3 5

Примечание

Пояснение к примеру:

10. Вычисление минимальной описанной окружности множества точек плоскости

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

На плоскости задано n точек. Требуется найти окружность минимального радиуса, содержащую все точки.

Формат входного файла

В первой строчке входного файла записано число n — количество точек. В последующих n строчках через пробел записаны пары вещественных чисел (x_i, y_i) , задающих координаты точек на плоскости.

Формат выходного файла

В выходной файл выведите три числа x, y, R — координаты центра и радиус минимальной окружности соответственно.

Примеры

standard input	standard output
5 2 2 -0.5 1 1 3 0.8 -0.2 3 1	1 1 2

11. Диаметр множества точек плоскости

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

Диаметром D множества точек X называется максимальное попарное расстояние между всеми различными точками этого множества:

$$D = \max_{a, b \in X} \text{distance}(a, b).$$

Формат входного файла

На первой строчке входного файла записано число n — количество точек в множестве. На последующих n строчках через пробел записаны пары вещественных чисел (x_i, y_i) , задающие координаты точек X_i на плоскости.

Формат выходного файла

В выходной файл выведите координаты двух точек, принадлежащих заданному множеству, расстояние между которыми равно диаметру. Если таких пар точек несколько, выведите любую.

Примеры

standard input	standard output
5 -1 1 2 2 3 0 0 -1 -2 3	-2 3 3 0
4 -2 -1 3 2 2 -2 -1 3	-1 3 2 -2

13. Триангуляция Делоне множества точек

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

На плоскости задано множество точек. Триангуляцией называется планарный граф, все внутренние области которого являются треугольниками. Триангуляцией Делоне множества точек S называют триангуляцию $DT(S)$, в которой:

- никакая точка a из S не содержится в окружности, описанной вокруг любого треугольника T из $DT(S)$
- точка a не является ни одной из вершин T .

Формат входного файла

На первой строчке входного файла записано натуральное число n — количество точек в множестве. На последующих n строчках через пробел записаны пары вещественных чисел (x_i, y_i) , задающих координаты точек на плоскости.

Формат выходного файла

В первой строчке выведите натуральное число p — число треугольников в триангуляции. На следующих p строчках выведите номера точек против часовой стрелки для каждого треугольника. Вершины нумеруются с единицы в порядке появления в исходных данных. Если триангуляций несколько, выведите любую из них.

Примеры

standard input	standard output
4 0 0 2 0 2 2 0 2	2 1 2 3 1 3 4
8 0 0 2 0 2 2 0 2 0.5 0.5 0.5 1.5 1.5 1.5 1.5 0.5	8 4 5 6 4 1 5 1 2 5 2 5 8 2 3 8 3 7 8 3 4 7 4 6 7

14. Триангуляция многоугольника

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

На плоскости задан произвольный многоугольник с дырками. Триангуляцией многоугольника P назовём его разбиение на множество попарно непересекающихся треугольников, объединение которых даёт P . В данной задаче требуется найти триангуляцию многоугольника.

Формат входного файла

На первой строчке входного файла записано натуральное число n – количество точек внешнего контура многоугольника. На последующих n строчках через пробел записаны пары вещественных чисел (x_i, y_i) , задающих координаты точек на плоскости. Вершины обходятся против часовой стрелки. На следующей строчке записано натуральное число m – количество дырок. Каждая дырка задаётся в формате: натуральное число k – количество точек в дырке и на последующих k строчках через пробел записаны пары вещественных чисел (x_i, y_i) , задающих координаты точек на плоскости. Вершины дырок обходятся по часовой стрелке.

Формат выходного файла

В первой строчке выведите натуральное число p – число треугольников в триангуляции. На следующих p строчках выведите номера вершин против часовой стрелки для каждого треугольника. Вершины нумеруются с единицы в порядке появления в исходных данных. Если триангуляций несколько, выведите любую из них.

Примеры

standard input	standard output
4 0 0 2 0 2 2 0 2 0	2 1 2 3 1 3 4
4 0 0 2 0 2 2 0 2 1 4 0.5 0.5 0.5 1.5 1.5 1.5 1.5 0.5	8 4 5 6 4 1 5 1 2 5 2 5 8 2 3 8 3 7 8 3 4 7 4 6 7

15. Пересечение множества отрезков

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

На плоскости задано множество отрезков. Требуется найти все точки пересечения. При наложении отрезков считать две точки, концы общей части этих отрезков. Каждую точку нужно задать отрезками проходящими через нее.

Формат входного файла

В первой строке задано целое число n — количество отрезков. Следующие n строк содержат по 4 вещественных числа — координаты начала и конца i -го отрезка.

Формат выходного файла

В первую строку выведите число m — количество точек пересечения отрезков. В следующие m строчек выведите количество отрезков образующих i -ую точку пересечения и номера этих отрезков. Отрезки нумеруются с единицы в порядке появления в исходных данных.

Примеры

standard input	standard output
4	2
4.0 4.0 11.0 11.0	3 1 2 3
1.0 9.0 15.0 9.0	2 2 4
6.0 12.0 14.0 4.0	
2.0 12.0 2.0 3.0	

16. Пересечение множества окружностей

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

На плоскости задано n окружностей. Требуется вывести все точки пересечения этих окружностей.

Формат входного файла

На первой строчке входного файла записано число n — количество окружностей. На последующих n строчках через пробел записаны три вещественных чисел x_i, y_i, R_i — координаты центра и радиус i -ой окружности соответственно.

Формат выходного файла

В выходной файл выведите число m — количество точек пересечения. На последующих m строчках через пробел выведите пары вещественных чисел (x_i, y_i) — координаты точек пересечения

Примеры

standard input	standard output
5	4
0 0 1	1 0
0 -3 2	0 -1
1 -1 1	1 -2
5 1 3	2 -1
-2 2 1	

17. Пересечение многоугольника с сеткой

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

На плоскости задан многоугольник. Требуется найти все точки пересечения многоугольника с прямыми вида $x = a$, $y = b$, где a, b — целые числа. При наложении ребра на прямую требуется вывести соответствующие ему вершины многоугольника.

Формат входного файла

В первой строчке входного файла записано натуральное число n — количество вершин многоугольника. В последующих n строчках через пробел записаны пары вещественных чисел (x_i, y_i) , задающих координаты вершин на плоскости. Вершины обходятся против часовой стрелки.

Формат выходного файла

В первой строчке выходного файла выведите целое неотрицательное число m — количество точек пересечения. В последующих m строчках выведите пары вещественных чисел (x_i, y_i) , задающих координаты пересечения многоугольника с прямыми. Точки обходятся в порядке обхода многоугольника.

Примеры

standard input	standard output
5 2 1 5 2 5 4 3 5 1 3	12 2 1 3 1 4 1 5 1 5 2 5 3 5 4 4 4.5 3 5 2 4 1 3 1.5 2
3 0.1 0.1 0.9 0.1 0.1 0.9	0

19. Ближайший сосед

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

На плоскости задано множество S , состоящее из n точек. Далее поступает m точек-запросов, для каждого запроса q_i выведите ближайшую к нему в евклидовой метрике точку $s \in S$.

Формат входного файла

На первой строчке входного файла записано натуральное число n — количество точек в множестве. На последующих n строчках через пробел записаны пары вещественных чисел (x_i, y_i) , задающие координаты точек s_i на плоскости.

На следующей строчке записано натуральное число m — количество запросов. На последующих m строчках через пробел записаны пары вещественных чисел (x_i, y_i) , задающие координаты точек q_i на плоскости.

Формат выходного файла

В выходной файл выведите m точек — ближайшую для каждого запроса точку. Если ближайших точек несколько, выведите любую.

Примеры

standard input	standard output
4 2 2 4 2 2 4 4 4 3 3 3 3 -100 4 4	2 2 4 2 4 4
3 0 2 1 2 2 1 2 -20 1 1 3	0 2 1 2

05. Массовая принадлежность точки выпуклому прямоугольнику

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

На плоскости задан планарный граф. Требуется на каждый запрос локализации выдать ячейку минимальной размерности, содержащую точку запроса.

Формат входного файла

В первой строчке входного файла записано три натуральных числа n_p , n_s и n_f — количество вершин, отрезков и граней графа соответственно. На последующих n_s строчках заданы вершины. Вершины задаются парами вещественных чисел (x_i, y_i) , задающих координаты точек. На следующих n_s строчках заданы отрезки. Каждый отрезок задается номерами двух вершин. На следующих n_f строчках задаются грани. Каждая грань задается количеством отрезков образующих её и номерами этих отрезков. На следующей строчке записано натуральное число m — количество запросов. На последующих m строчках следуют пары вещественных чисел (x_i, y_i) , задающих координаты точек-запросов. Объекты нумеруются с 0 в порядке появления во входном файле.

Формат выходного файла

В выходной файл выведите два натуральных числа: размерность ячейки и ее номер. Если точка лежит во внешней области вывести -1.

Примеры

standard input	standard output
6 11 6	1 4
10 10	2 1
10 -10	0 3
-10 10	-1
-10 -10	
5 - 5	
-5 5	
0 1	
1 3	
3 4	
0 4	
4 5	
3 5	
5 0	
2 5	
2 3	
2 0	
4 1	
3 7 9 6	
3 3 6 4	
3 7 5 8	
3 2 4 5	
3 1 2 10	
3 0 3 10	
4	
0 0	
5 5	
-10 -10	
100 100	

22. Кратчайший путь материальной точки между точками плоскости с полигональными препятствиями

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

На плоскости заданы две точки: начальное и конечное положение робота. Робот является материальной точкой. Также на плоскости заданы полигональные препятствия. Робот должен добраться из начальной точки в конечную по кратчайшему пути, при этом он может касаться препятствий. Кратчайший путь нужно задать точками, где робот меняет направление своего движения. Гарантируется существования пути и то, что робот изначально находится вне препятствия.

Формат входного файла

В первой строке заданы координаты старта и финиша соответственно. Во второй строке задано количество препятствий n . Далее следует n блоков, задающих препятствия. В первой строке каждого блока записано количество вершин m в препятствии. В следующих m строках заданы координаты вершин препятствия.

Формат выходного файла

В первой строке выходного файла выведите количество поворотов, сделанных роботом – k . В следующие k строчек выведите координаты точек пути, где робот совершал поворот. Точки выводятся в порядке обхода. Если кратчайших путей несколько, выведите любой из них.

Примеры

standard input	standard output
0.0 0.0 25.0 17.0	2
2	7.0 3.0
3	14.0 12.0
3.0 6.0	
7.0 3.0	
10.0 8.0	
3	
21.0 7.0	
20.0 14.0	
14.0 12.0	

Примечание

Пояснение к примеру:

example.pdf

31. Предикат «поворот»

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

Предикат «поворот» — геометрический предикат, принимающий три точки a , b и c на плоскости и определяющийся следующим образом:

$$\text{turn}(a, b, c) = \begin{cases} 1, & \text{если } c \text{ лежит слева от прямой, заданной точками } a \text{ и } b \\ 0, & \text{если } c \text{ лежит на прямой, заданной точками } a \text{ и } b \\ -1, & \text{если } c \text{ лежит справа от прямой, заданной точками } a \text{ и } b \end{cases}$$

Формат входного файла

В первой строке входного файла задано натуральное число n — количество тестов.

В следующих $3n$ строках входного файла заданы пары вещественных чисел — координаты точек a , b и c . Гарантируется, что точки a и b не совпадают.

Примечание: во всех тестах на корректность $n = 1$. Мультитест сделан для удобства тестирования на производительность.

Формат выходного файла

В выходной файл через пробел выведите n чисел — значений предиката для заданных точек.

Примеры

standard input	standard output
1 2 1.5 0.5 0.5 3.5 2.5	0
standard input	standard output
1 2 1.5 0.5 0.5 1.234 0.1123	1
standard input	standard output
1 2 1.5 0.5 0.5 5.21341 324124.5	-1