# Задача A. Прямые и точки

| | |
|---|---|
| Имя входного файла: | `lnp.in` |
| Имя выходного файла: | `lnp.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

Даны $n$ точек на плоскости; назовём их точками уровня 0.

Построим прямые, проходящие через все пары различных (в геометрическом смысле) точек уровня 0; назовём их прямыми уровня 1.

Рассмотрим точки, являющиеся пересечениями всех пар различных (в геометрическом смысле) прямых уровня 1; назовём их точками уровня 2.

Найдите прямую, проходящую через хотя бы одну точку уровня 0 и содержащую как можно больше точек уровня 2. Точки считаются с учётом кратности: если несколько точек уровня 2 совпадают, то считается всё равно каждая из них.

Обратите внимание, что среди объектов каждого уровня могут встречаться одинаковые.

## Формат входного файла

В первой строке входного файла задано число $n$ ($1 \leqslant n \leqslant 10$). Следующие $n$ строк содержат по два числа $x_i$ $y_i$ каждая через пробел — координаты $i$-ой точки уровня 0 ($-10 \leqslant x_i, y_i \leqslant 10$). Все числа во входном файле целые.

## Формат выходного файла

В первой строке выходного файла выведите $k$ — число точек уровня 2 на найденной прямой. Во второй строке выведите три числа $a$, $b$ и $c$ через пробел — координаты прямой, задающие уравнение прямой $ax + by + c = 0$. Числа $a$, $b$ и $c$ должны быть целыми и не превышать по модулю $10^9$; кроме того, хотя бы одно из чисел $a$ и $b$ должно быть отлично от нуля. Если ответов с максимальным $k$ несколько, разрешается выводить любой из них.

## Примеры

| lnp.in | lnp.out |
|---|---|
| 4<br>0 0<br>0 1<br>1 0<br>1 1 | 7<br>1 -1 0 |
| 3<br>-5 -5<br>5 5<br>-5 -5 | 0<br>1 1 10 |

В первом примере на прямой $x = y$ семь точек; точки $(0, 0)$ и $(1, 1)$ считаются с кратностью 3 каждая, а точка $(0.5, 0.5)$ — с кратностью 1.

Во втором примере прямых первого уровня — две: одна проходит через первую и вторую точки, а другая — через вторую и третью. Поскольку в геометрическом смысле эти две прямые совпадают (эти прямые даже проведены через одинаковые в геометрическом смысле пары точек), множество точек второго уровня пусто, и поэтому ответом является любая прямая, проходящая через хотя бы одну из точек уровня 0.

# Задача B. Выпуклая оболочка

| | |
|---|---|
| Имя входного файла: | `convex.in` |
| Имя выходного файла: | `convex.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

*Выпуклая оболочка* множества точек на плоскости — это выпуклый многоугольник минимальной площади такой, что все точки множества лежат у него внутри или на границе. Известно, что для любого множества точек выпуклая оболочка единственна с точностью до порядка обхода её вершин.

По данному множеству из $n$ точек постройте его выпуклую оболочку.

## Формат входного файла

В первой строке входного файла задано число $n$ ($3 \leqslant n \leqslant 100$). Следующие $n$ строк содержат по два числа $x_i$ $y_i$ каждая через пробел — координаты $i$-ой точки ($-1000 \leqslant x_i, y_i \leqslant 1000$). Все числа во входном файле целые.

## Формат выходного файла

В первой строке выходного файла выведите $k$ — количество вершин выпуклой оболочки. В следующих $k$ строках выведите по два числа $x'_k$ $y'_k$ через пробел — координаты вершин многоугольника, являющегося выпуклой оболочкой, в порядке их обхода против часовой стрелки. Каждый из углов многоугольника должен быть строго больше нуля и строго меньше 180 градусов.

Гарантируется, что точки во входном файле таковы, что такой многоугольник существует.

## Примеры

| convex.in | convex.out |
|---|---|
| 3<br>0 0<br>1 1<br>0 1 | 3<br>0 0<br>0 1<br>1 1 |
| 6<br>0 0<br>0 2<br>2 0<br>2 2<br>2 1<br>1 1 | 4<br>0 2<br>2 2<br>2 0<br>0 0 |

В первом примере надо всего лишь расположить три данные точки в порядке обхода против часовой стрелки.

Во втором примере надо исключить те точки, которые лежат внутри или на границе квадрата $2 \times 2$.

# Задача C. Двухкратная подстрока

| | |
|---|---|
| Имя входного файла: | `double.in` |
| Имя выходного файла: | `double.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

Дана строка $S$ длины $n$ и число $k$. Найдите в строке $S$ такую подстроку длины $k$, которая встречается в ней по крайней мере два раза, или выясните, что такой подстроки нет.

## Формат входного файла

В первой строке входного файла задана строка $S$; её длина $n$ — от 1 до 100 000 символов, включительно. Во второй строке задано целое число $k$ ($1 \leqslant k \leqslant n$). Строка состоит только из маленьких букв латинского алфавита.

## Формат выходного файла

Если подстроки длины $k$, встречающейся хотя бы два раза, не существует, выведите слово "`NONE`" в первой строке выходного файла. В противном случае выведите любую из таких подстрок в первой строке выходного файла.

## Примеры

| double.in | double.out |
|---|---|
| ast<br>1 | NONE |
| blinkingblueblogger<br>2 | in |
| aaaaaab<br>5 | aaaaa |

В первом примере в строке нет одинаковых букв, поэтому следует вывести "`NONE`".

Во втором примере две подходящие подстроки — "`bl`" и "`in`"; можно вывести любую из них.

В третьем примере вхождения подстроки "`aaaaa`" перекрываются.