

Problem I. Friends and Berries - 2

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

There is a group of n children. According to a proverb, every man to his own taste. So the children value strawberries and raspberries differently. Let us say that i -th child rates his attachment to strawberry as s_i and his attachment to raspberry as r_i .

According to another proverb, opposites attract. Surprisingly, those children become friends whose tastes differ.

Let us define friendliness between two children v and u as

$$p(u, v) = (s_u - s_v)^2 + (r_u - r_v)^2.$$

The friendliness between three children v , u , w is half the sum of pairwise friendlinesses:

$$p(u, v, w) = \frac{p(u, v) + p(u, w) + p(v, w)}{2}.$$

The best friends are such pairs of children (u, v) that $u \neq v$ and $p(u, v) \geq p(u, v, w)$ for every w . Your goal is to find all pairs of best friends.

Input

In the first line there is one integer n , the number of children ($2 \leq n \leq 2 \cdot 10^5$).

Each of the next n lines contains two integers s_i and r_i ($-10^8 \leq s_i, r_i \leq 10^8$).

It is guaranteed that, for every two children, their tastes differ. In other words, if $u \neq v$, then $s_u \neq s_v$ or $r_u \neq r_v$.

Output

On the first line, output the number of pairs of best friends.

After that, output those pairs. Each pair should be printed on a separate line. A pair is denoted by two integers: the indices of children in this pair. Children are numbered in the order of input starting from 1. You can output pairs in any order. You can output indices in each pair in any order.

It is guaranteed that the required number of pairs doesn't exceed 10^6 .

Examples

standard input	standard output
4 0 0 1 0 0 1 1 1	2 1 4 2 3
3 0 0 0 10 5 8	0