

CIRCLES (100 points)

You are working on the collision system for a large infantry battle simulation. Performing a precise collision check between a pair of units is computationally expensive, so your job is to implement a preliminary pass which eliminates units that cannot possibly be in contact. You can treat the battlefield as a 2-dimensional plane, with each unit having a bounding circle you should use to detect potential collisions (all bounding circles have the same radius R). All coordinates in this plane will be given as integers.

Two bounding circles are considered to be colliding if they overlap or touch – in other words, if the Euclidean distance between their centres is $\leq 2R$. Fig 1 shows a pair of touching circles that are considered colliding.

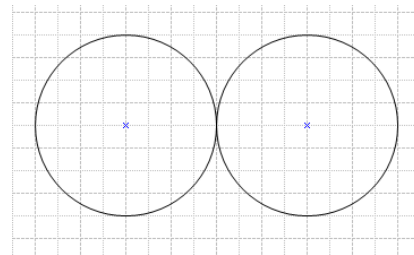


Fig 1

Because the unit positions went through a collision-resolving pass in the previous game frame, and because the units are sufficiently slow-moving, you can assume that no unit centre is inside another unit's bounding circle. The situation in figure 2 will never occur.

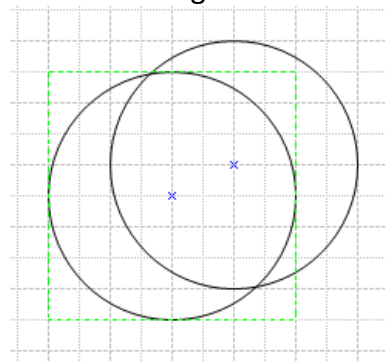


Fig 2

TASK

Write a program that, given a set of N unit positions $P_i = (x_i, y_i)$ and the unit radius R , computes the number of colliding unit pairs. A unit pair (i, j) is colliding if and only if

$$(x_i - x_j)^2 + (y_i - y_j)^2 \leq (2r)^2$$

Pairs (i, j) and (j, i) are entirely equivalent: do not count them twice.

CONSTRAINTS

$$2 \leq N \leq 100,000$$

$$1 \leq R \leq 10$$

$$0 \leq x_i \leq 32767$$

$$0 \leq y_i \leq 32767$$

No unit pairs are within distance R of one another:

$$(x_i - x_j)^2 + (y_i - y_j)^2 \geq r^2 \quad \text{for all } i, j$$

TIME LIMIT

A time limit of 3 seconds will be placed upon your programme for each test case.

INPUT

Your programme must read from standard input, the following set of data:

- Line 1 contains an integer N , the number of units. Units are numbered from 0 to $N - 1$ inclusive.
- Line 2 contains an integer R , the radius of each collision circle
- Each of the next N lines gives a unit position using two integers x and y , separated by a single space.

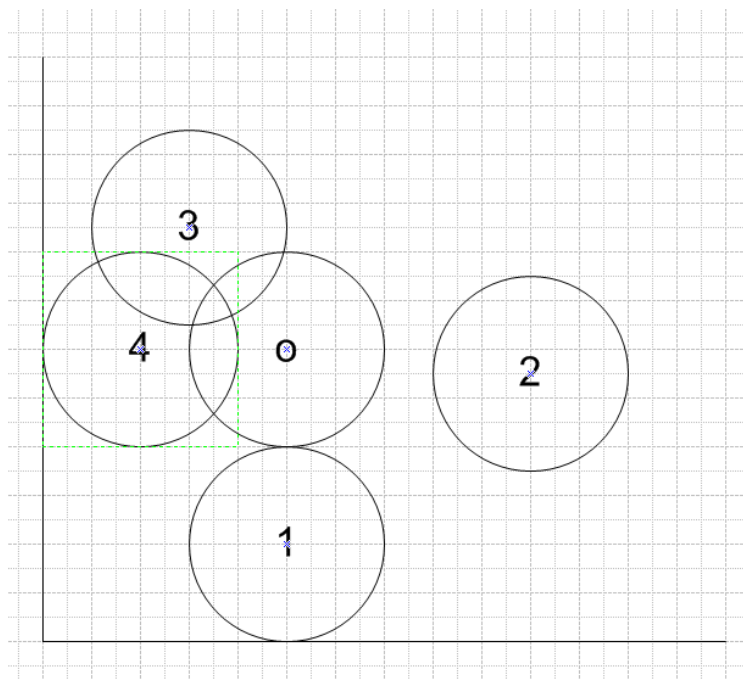
OUTPUT

Your programme must write to standard output a single integer M , the number of unique colliding pairs of units.

EXAMPLE

Sample input	Sample output
5 4 10 12 10 4 20 11 6 17 4 12	4

This example is shown below:



This work is licenced under a [Creative Commons Licence](https://creativecommons.org/licenses/by-nc/4.0/).