

D. Looking for Owls

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Emperor Palpatine loves owls very much. The emperor has some blueprints with the new Death Star, the blueprints contain n distinct segments and m distinct circles. We will consider the segments indexed from 1 to n in some way and the circles — indexed from 1 to m in some way.

Palpatine defines an owl as a set of a pair of distinct circles (i, j) ($i < j$) and one segment k , such that:

1. circles i and j are symmetrical relatively to the straight line containing segment k ;
2. circles i and j don't have any common points;
3. circles i and j have the same radius;
4. segment k intersects the segment that connects the centers of circles i and j .

Help Palpatine, count the number of distinct owls on the picture.

Input

The first line contains two integers — n and m ($1 \leq n \leq 3 \cdot 10^5$, $2 \leq m \leq 1500$).

The next n lines contain four integers each, x_1, y_1, x_2, y_2 — the coordinates of the two endpoints of the segment. It's guaranteed that each segment has positive length.

The next m lines contain three integers each, x_i, y_i, r_i — the coordinates of the center and the radius of the i -th circle. All coordinates are integers of at most 10^4 in their absolute value. The radius is a positive integer of at most 10^4 .

It is guaranteed that all segments and all circles are distinct.

Output

Print a single number — the answer to the problem.

Please, do not use the `%lld` specifier to output 64-bit integers in C++. It is preferred to use the `cout` stream or the `%I64d` specifier.

Examples

input
1 2 3 2 3 -2 0 0 2 6 0 2
output
1

input
3 2 0 0 0 1 0 -1 0 1 0 -1 0 0 2 0 1 -2 0 1
output
3

input

1 2
-1 0 1 0
-100 0 1
100 0 1

output

0

Note

Here's an owl from the first sample. The owl is sitting and waiting for you to count it.