

Click the Circle

Input file: **standard input**
Output file: **standard output**
Time limit: 5 seconds
Memory limit: 1024 megabytes

This is a modified version of the game osu!.

There are two types of objects:

1. Circle: c_x, c_y, t :

- The center of this circle is at (c_x, c_y) , and its hit time is t .
- The circle's presence in the game is during the time interval $[t - d, t + d]$.
- All circles share the same radius, r .



2. Slide: s_x, s_y, t_x, t_y, u, v :

A slide contains two parts:

- (a) A moving circle, which also has a radius of r .
- (b) A frame that holds the path of this circle.

A slide holds the path of a moving circle, whose radius is always r .

The move can be described as follow:

- (a) At time $u - d$, the circle appears with the frame, and the center of the circle is (s_x, s_y)
- (b) At time u , the center of the circle starts moving towards (t_x, t_y) at a constant speed.
- (c) At time v , the center of the circle reaches (t_x, t_y) .
- (d) After time $v + d$, the circle and the frame disappear.



Two objects are considered to intersect if, at a certain time t , both are present and their shapes overlap (boundaries inclusive).

The two components of a slide should be treated as **two** distinct objects.

Given the values r , d , and n objects, calculate the number of intersecting pairs of objects.

Input

The first line of the input contains three positive integers n, r, d ($1 \leq n, r, d \leq 10^3$).

The next n lines describe the objects. Each line starts with a number $type \in \{1, 2\}$, indicating the type of the object.

- If $type = 1$, then there will be three integers c_x, c_y, t , indicating a circle.
- If $type = 2$, then there will be six integers s_x, s_y, t_x, t_y, u, v , indicating a slide.

It is guaranteed that $1 \leq c_x, c_y, s_x, s_y, t_x, t_y \leq 10^4$, $1 \leq t, u, v \leq 10^3$ and $u < v$.

Output

Output a single line contains a single integer, indicating the number of pairs.

Examples

standard input	standard output
2 1 1 1 1 1 2 1 2 2 3	1
2 1 1 1 1 1 2 1 3 2 3	0
2 1 1 1 3 3 2 2 5 5 5 1 2 4	3
2 1 1 2 1 1 1 5 2 4 2 5 5 5 1 2 4	2
2 1 1 2 10 1 10 20 2 4 2 1 10 20 10 2 4	6