E. Playing the ball

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

A coder cannot sit and code all day. Sometimes it is a good idea to rise from the desk, have a rest, have small talk with colleagues and even play. The coders of the F company have their favorite ball game.

Let's imagine the game on the plane with a cartesian coordinate system. The point (0, 0) contains the player who chooses an arbitrary direction and throws a ball in that direction. The ball hits the plane at distance d from the player's original position and continues flying in the same direction. After the ball hits the plane for the first time, it flies on and hits the plane again at distance $2 \cdot d$ from the player's original position and so on (it continue flying in the chosen direction and hitting the plane after each d units). All coders in the F company are strong, so the ball flies infinitely far away.

The plane has n circles painted on it. If a ball hits the plane and hits a circle that is painted on the plane (including its border), then the player gets one point. The ball can hit multiple circles at once and get one point for each of them (if the ball hits some circle x times during the move, the player also gets x points). Count the maximum number of points a player can get if he throws a ball in the arbitrary direction. Note that the direction may have real cooridinates.

Input

The first line contains two space-separated integers — n и d $(1 \le n \le 2 \cdot 10^4;\ 5 \le d \le 10)$. Next n lines contain the circles' description. The i-th line contains three space-separated integers x_i, y_i, r_i $(- 10000 \le x_i, y_i \le 10000;\ 1 \le r \le 50)$, where (x_i, y_i, r_i) are the coordinates of the center and the radius of the circle, correspondingly. The point (0, 0) is not inside or on the border of some circle.

Output

Print a single integer — the maximum number of points you can get.

Examples

input

```
2 5
1 1 1
5 0 1
```

output

```
1
```

input

```
2 5
4 0 3
5 3 1
```

output

```
2
```

input

```
1 10
20 0 10
```

output

```
3
```