Bring Your Own Bombs

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2 seconds
Memory limit:	256 megabytes

A sabotage platoon has completed its mission and planted some bombs on an enemy territory. Now they are about to report to their commanders how many enemy units will be destroyed. They have assigned this task to you.

The enemy territory is represented by a tile grid where the enemy platoons are located. Enemy units are standing in platoons that are represented by accurate rectangles, and they do not intersect each other. Each tile contains at most one enemy unit. Bombs have awesome firepower that can destroy all enemy units on the same tile line with the bomb: either horizontal or vertical, but not both. Unfortunately, all bombs are created manually by inexperienced engineers, and therefore, it is very hard to say how and if they explode or not. Only two values are known for certain for each bomb: p_1, the probability that the bomb will explode horizontally, and p_2, the probability that the bomb will explode vertically. It is implicitly known that with probability of $100\% - p_1 - p_2$ the bomb will not explode at all.

The sabotage platoon only plants the bombs outside the enemy platoons. Given the description of the enemy territory and planted bombs, you are to calculate the expected number of destroyed enemy units.

Input

The first line contains two integers N and M: the number of enemy platoons and bombs respectively ($1 \le N, M \le 10^5$).

The next N lines of input contain the descriptions of enemy platoons, one per line. Each platoon is described by four integers x_1, y_1, x_2, y_2: the coordinates of the tiles at the opposite corners of the rectangle ($x_1 \le x_2$, $y_1 \le y_2$).

Then the next M lines of input contain the description of bombs, one per line. Each bomb is described by integers x, y and numbers p_1, p_2: the coordinates of the tiles with bombs and the quality characteristics of the bombs given as an non-negative integer percentage respectively ($p_1 + p_2 \le 100$).

All coordinates do not exceed 10^9 by absolute value.

Output

Output must contain a single real number: the answer to the problem with absolute or relative error no more than 10^{-9}.

Examples

stdin	stdout
1 1 -1 -1 1 1 0 3 33 33	0.990000000000000
1 2 1 1 5 5 0 2 100 0 2 0 0 100	9.000000000000000
2 2 0 3 2 5 3 0 5 2 1 1 30 60 4 4 50 40	4.980000000000000