# Help Mayor Editorial

There are $O(R)$ lines that intersects with the circle and that can be represented as $x = c$ (where $c$ is an integer).
For each of them, we can apply Pythagorean theorem to find the range of $y$ coordinates of the intersection of the line and the circle, so we can find the number of lattice points that is within the circle and on the line $x = c$.

However, the floating-point error is a big issue here.
For example, given $(X, Y) = (-0.0001, 0)$, $R = 100000$, when calculating the upper bound of the intersection of line $x = 0$ and the circle, we have to calculate $(10^5)^2 - (10^{-4})^2$, which requires far more precision than double type.
To resolve the issue, multiply $X, Y, R$ by $10^4$ to make them integers. Then, we can count the number of points where both $x$ and $y$ are multiples of $10^4$.
Just like before, there are $O(R)$ lines that intersects with the circle and that can be represented as $x = 10^4 c$ (where $cc$ is an integer), so it can be solved in the similar way. Since entire calculation are done with integer, we don’t have to care about precision errors.

The complexity is $O(R\log(R))$ in total, with assumption that calculating the square root requires an $O(\log(R))$ time.
Since $X, Y$ and $R$ cannot be inputted precisely in most cases, just multiplying them by $10^4$may not precisely results in integers. Therefore, use proper functions to round into integers, such as round function in C++. (In C++, a double value is rounded to int value in the direction of $00$, so an attempt to convert to int after adding $0.5$ fails is it is negative)