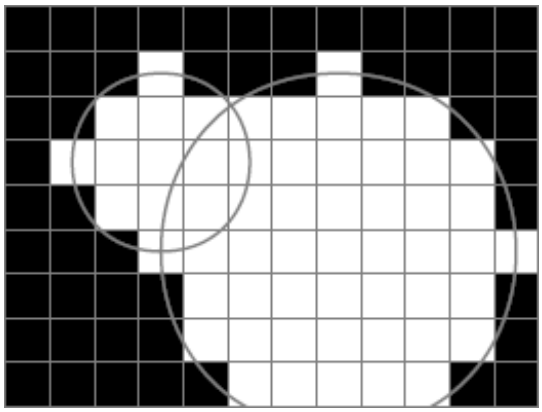
Circles On A Screen

Yesterday Andrew wrote a program that draws n white circles on a black screen. The screen is monochrome and it has a resolution w x h pixels. Pixels are numbered from upper left corner (0, 0) to bottom right one (w-1, h-1).

A circle with the center at pixel (x_c, y_c) and the radius r consists of the pixels with coordinates (x, y) such that $(x_c-x)^2 + (y_c-y)^2 <= r^2$. If the circle does not fit on the screen, it is truncated. If some pixel belongs to two or more circles, it is white.



The resulting picture was very nice, so Andrew decided to copy it to his wall. He has white wallpaper and he can only draw some parts of wall into black. Now he wants to know the amount of paint he needs.

He copies the picture exactly pixel-to-pixel, so you should write a program that calculates the number of black pixels left on a screen after drawing n circles.

Input

In the first line of input file there is an integer T, the number of test cases. T test cases follow.

Each test case begins with a line where there are three integers: w, h, and n (1 <= w, h <= 20.000; 1 <= n <= 100). Each of the following n lines contains descriptions of the circles. In i+1-th line there are three integers: x_i, y_i, r_i (0 <= x_i < w; 0 <= y_i < h; 0 <= r_i <= 40 000). They denote a circle with the center at pixel (x_i, y_i) and radius r_i.

Output

For each test case you should output exactly one number - the number of black pixels left on the screen.

Example

Input:
2
5 3 2
1 1 1
3 1 1
12 9 2
3 3 2
7 5 4

Output:
6
51

Note: The picture corresponds to the second test case of the example.