E Clocks

Problem

During the past few years Tim E. has been collecting circle-shaped clocks and he exposed them on a big wall in his living room. He wants to buy another big clock, but because his wall is almost full he wants to know what the radius of the biggest clock is that could be exposed on the wall without overlapping any other clock.

The wall is modelled as a rectangle with width W and height H. For each clock the coordinates (x_i, y_i) of the centre-point and the radius r_i are given. The clocks will not overlap with the boundaries of the wall, nor will two clocks overlap, however they might touch each other.

Input

The first line of the input contains a single number: the number of test cases to follow. Each test case has the following format:

- One line with two integers W and H satisfying $1 \leq W, H \leq 1,000,000$: the width and the height of the wall, respectively.
- One line with an integer C satisfying $0 \leq C \leq 50$: the number of clocks.
- C lines, each with three integers x_i, y_i and r_i satisfying $r_i > 0$, $0 \leq x_i - r_i$, $x_i + r_i \leq W$, $0 \leq y_i - r_i$ and $y_i + r_i \leq H$: the location and radius of an existing clock.

Integers on the same line are separated by single spaces. For any pair of clocks i and j ($i \neq j$), $(x_i - x_j)^2 + (y_i - y_j)^2 \geq (r_i + r_j)^2$.

Output

For every test case in the input, the output should contain a single real number, on a single line: the maximum radius of a clock that fits on the wall without overlapping with any existing clock or the boundaries of the wall.

Your answer should have either an absolute or a relative error of at most 10^{-6}. Make sure that you print enough decimals. In particular, if you program in C++, do not use plain `cout << M << endl;` to output your answer (a double) M, because this sometimes produces too few decimals. Instead, you may use `printf ("%lf\n", M);` .

Example

Input	Output
1 10 10 4 2 2 1 2 8 1 8 2 1 8 8 1	3.242640687119285

This page intentionally left blank.