

There are N circles. Each circle can move 1 unit distance per unit time (here distance is Euclidean distance). A circle moved d unit distance means its center moved a path (not necessarily in straight line) of d unit distance. When the center moves its body also moves with it (by body we mean circumference of course!). The circles can overlap each other. Our target is to provide all the circles enough time so that they can move themselves to such positions that all of their circumferences go through a common point. Find out the minimum time you need to provide them.

For example, suppose there are two circles one at co-ordinate $(0, 0)$ and another at co-ordinate $(10, 0)$. Both of them have radius 1. After 4 unit time first circle can move to $(4, 0)$ and the second one to $(6, 0)$. Both of them go through the point $(5, 0)$.

Input

First line of the test file contains a positive integer T ($T \leq 30$), number of test cases. Hence follow T test cases, each starting with a positive integer N ($1 < N \leq 100$). Next N lines describe the circles of this test case. Each of these N lines contains three numbers $x y r$. (x, y) is the center and r is the radius of the circle. You may assume that x and y are integers having absolute value not greater than 1000. r may be floating point number having at most 6 digits after its decimal point and it will not be greater than 1000. No two circles of the input will be equal.

Output

For each test case, output the minimum time.

Errors up to $5 * 10^{-4}$ will be ignored. It is safe for you to output at least four digits after the decimal point.

Sample Input

```
2
2
0 0 1
10 0 1
2
1 0 10
-1 0 10
```

Sample Output

```
4.0000
0.0000
```