Missile Interception

Time Limit: 2000/1000 MS (Java/Others) Memory Limit: 65536/65536 K (Java/Others)
Total Submission(s): 141 Accepted Submission(s): 47

Problem Description

A country's defense department has been informed there are n enemy's missile silos in an area. To avoid potential threat, DOD plans to build up a missile defense base to ensure any enemy's missiles can be shot down once they have been launched. Due to a tight budget, only one defense base can be built(which means all defense missiles can only be launched at one position). DOD knows the best strategy is to minimize the longest time required to intercept any enemy's missiles. The good news is that DOD has got a confidential document which reports each enemy's missiles' parameter, including their launching coordinate, direction and speed. DOD is free to launch its defense missile in any direction, but only AFTER enemy's missile has launched (or serious diplomatic crisis could occur).
To simply the problem, we regard the area as a 2D plane. A missile can only fly alone a straight line with a fixed speed (its initial speed) and never comes down, until it successfully intercept or being intercepted by another missile. Now DOD wants to know with the best strategy described above, what's the longest time required to intercept any enemy's missile.
Note again, DOD doesn't need to intercept enemy's missiles as soon as they have been launched. The only requirement is not to launch the defense missile before enemy's missile.

Input

Multiple test cases (not exceed 5).
For each test case:
• The first line contains two numbers n,V($2 \le n, V <= 100$), indicating the number of enemy's missiles and the speed of all defense missiles;
• Then n lines follow. The ith line has 5 integers $x_i, y_i, dx_i, dy_i, v_i$, which means a enemy's missile can be launched at coordinate (x_i, y_i) with direction vector (dx_i, dy_i), and its speed is v_i.($0 \le |x_i|, |y_i|, |dx_i|, |dy_i| \le 100, 1 \le v_i \le 100$)
It's guaranteed $V > v_{max}$.

Output

For each test case, output one line indicating the longest time mentioned above. Keep 4 decimal places.

Sample Input

```
2 2
2 0 3 0 1
0 0 3 1 1
2 2
0 0 0 1 1
0 2 0 -1 1
```

Sample Output

```
0.5081
0.3333
```

Source

2017 Multi-University Training Contest - Team 9

Recommend

liuyiding | We have carefully selected several similar problems for you: 6170 6169 6168 6167 6166