I: Moving Points

Consider a number of Target points in a plane. Each Target point moves in a straight line at a constant speed, and do not change direction. Now, consider a Chaser point that starts at the origin, and moves at a speed faster than any of the Target points. The Chaser point moves at a constant speed, but it is capable of changing direction at will. It will 'Catch' a Target point, and then move from there to catch another Target point, and so on.

Given the parameters of the Chaser point and the Target points, what is the least amount of time it takes the Chaser point to catch all of the Target points? 'Catch' simply means that the Catcher and the Target occupy the same point in the plane at the same time. This can be instantaneous; there's no need for the Catcher to stay with the Target for any non-zero length of time.

Input

There will be several test cases in the input. Each test case will begin with two integers

N C

Where **N** ($1 \le$ **N** ≤ 15) is the number of Target points, and **C** ($0 <$ **C** $\le 1,000$) is the speed of the Chaser point. Each of the next **N** lines will have four integers, describing a Target point:

X Y D S

Where (**X**,**Y**) is the location in the plane ($-1000 \le$ **X**,**Y** $\le 1,000$) of that Target point at time 0, **D** ($0 \le$ **D** < 360) is the direction of movement in Degrees (0 degrees is the positive X axis, 90 degrees is the positive Y axis), and **S** ($0 \le$ **S** $<$ **C**) is the speed of that Target point. It is assumed that all Target points start moving immediately at time 0.

The input will end with a line with two 0s.

Output

For each test case, output a single real number on its own line, representing the least amount of time needed for the Chaser point to catch all of the Target points. Print this number to exactly 2 decimal places, rounded. Output no extra spaces, and do not separate answers with blank lines.

Sample Input

```
2 25
19 19 32 10
6 45 133 19
5 10
10 20 45 3
30 10 135 4
100 100 219 5
10 100 301 4
30 30 5 3
0 0
```

Sample Output

```
12.62
12.54
```