

## Problem E. Archer Vlad

Input file: *standard input*  
Output file: *standard output*  
Time limit: 3 seconds  
Memory limit: 512 mebibytes

Vlad was an exemplary student known for his exceptional adventures, many of which have been preserved as tasks for programming contests. But such a restless life had exhausted Vlad a bit too much. “Wherever I go, there are only problems! I’m done!” – he announced, right before he left the university, headed toward Bieszczady mountains.

Vlad rented a small hut, in which he spent the first few months of his vacations. But soon boredom started having its grasp on him, so Vlad decided to find himself a hobby: he bought a bow, a few arrows and started his daily practice of archery. And after a few more months under solid training, Vlad reached some very satisfying results, as he was able to shoot an arrow with an astonishing speed of  $C$  meters per second. But it was hard to enjoy such accomplishments with no one around.

“Check this out! I’m gonna stand right here and shoot an arrow so quick, that it’s gonna fly over each and every single one of all those trees!” — Vlad exclaimed to you, a young programmer who decided to pay him a visit. Vlad tightened the bow and shot the first arrow. Its feathers were swaying in the air, its arrowhead was shining in the sky... but it hit a tree. “Hold on, let me try again!”

His second attempt was even more spectacular than the first one. But this arrow could not find its way out of the forest either. “One last time!” Vlad shouted, reaching his hand to the sack once again. Then you stopped him. Afraid that Vlad would run out of arrows, you decided to find an optimal angle at which he should aim. And so you reached to the computer in your backpack, ready to solve this problem in the UJ TCS style.

Vlad stands on a Cartesian plane at the point  $(0,0)$ . Both points  $(0,1)$  and  $(1,0)$  are precisely 1 meter away from Vlad. There are  $N$  trees numbered from 1 to  $N$ , and the tree number  $i$  is represented by a vertical segment connecting the points  $(x_i, 0)$  and  $(x_i, y_i)$  for some positive integers  $x_i$  and  $y_i$ . When Vlad shoots at an angle  $\alpha$  it gives his arrow an initial horizontal speed  $v_x$  equal to  $C \cdot \cos(\alpha)$  and an initial vertical speed  $v_y = C \cdot \sin(\alpha)$ . The arrow is not affected by air resistance and its trajectory is a parabola (to be precise, its horizontal speed  $v_x$  stays constant throughout the whole flight, while  $v_y$  decreases linearly with per-second loss equal to  $g$ ), containing the point  $(0,0)$ . We assume that the acceleration due to gravity is  $g = 10m/s^2$ . Vlad’s objective will be reached if the trajectory of the arrow he shot does not intersect any of the trees (or more specifically intervals representing them) at any point. Furthermore, the trajectory of the arrow must intersect the x-axis at the point which has a greater x-coordinate than any tree.

Output a possible value of  $\tan(\alpha)$  which allows Vlad to meet these conditions.

### Input

The first line of input contains the number of test cases  $z$ . The descriptions of the test cases follow.

The first line of each case consists of an integer  $1 \leq C \leq 10^9$  which is the speed of Vlad’s arrow in meters/second.

The second line of each case contains a single integer  $1 \leq N \leq 100\,000$  – the number of trees.

For each case the next  $N$  lines contain two integers  $x_i, y_i$  ( $1 \leq x_i, y_i \leq 10^9$ ) each. The  $i$ -th tree is represented by a vertical segment between points  $(x_i, 0)$  and  $(x_i, y_i)$ .

The sum of  $N$  in all the test cases does not exceed 300 000.

### Output

For each case output a single number with exactly 3 digits after the decimal point. It must approximate one of the correct values of  $\tan(\alpha)$  with error no greater than  $10^{-3}$ . You may assume that the solutions always exist, and that any correct value of  $\tan(\alpha)$  is contained in an interval of solutions of length at least  $10^{-2}$ .

## Example

standard input	standard output
3	2.000
5	3.000
1	2.429
1 1	
5	
1	
1 1	
13	
1	
7 7	