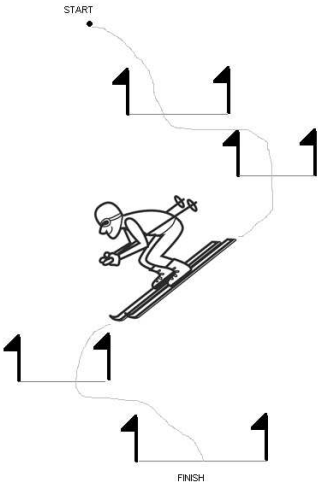
G
Slalom



In spite of the scarcity of snowfall in Madrid, interest in winter sports is growing in the city, especially with regard to skiing. Many people spend several weekends or even full weeks improving their skills in the mountains.

In this problem we deal with only one of the multiple alpine skiing disciplines: slalom. A course is constructed by laying out a series of gates, which are formed by two poles. The skier must pass between the two poles forming each gate. The winner is the skier who takes the least time to complete the course while not missing any of the gates.

You have recently started to learn to ski, but you have already set yourself the goal of taking part in the Winter Olympic Games of 2018, for which Madrid will presumably present a candidature. As part of the theoretical training, you need to write a program that calculates, given a starting point and a series of gates, the minimum-length path starting from the point given and passing through each gate until you reach the last one, which is the finish line. You may assume that the gates are horizontal and are ordered from highest to lowest, so that you need to pass through them in order. You consider yourself an accomplished skier, so you can make any series of turns, no matter how difficult, and your only concern is minimizing the total length of the path.

Input

The first line of each case gives the number of gates n ($1 \leq n \leq 1\,000$). The next line contains two floating point numbers, the Cartesian coordinates x and y of the starting position, in that order. Next come n lines with three floating point numbers each, y x_1 x_2, meaning that the next gate is a horizontal line from (x_1, y) to (x_2, y). You can safely assume that $x_1 < x_2$. The values of y are strictly decreasing and are always smaller than that of the starting position. The last gate represents the finish line. All coordinates are between $-500\,000$ and $500\,000$, inclusive. A value of 0 for n means the end of the input. A blank line follows each case.

Output

For each test case, output a line with the minimum distance needed to reach the finish line. Your answer should be accurate to within an absolute or relative error of 10^{-7}.

Sample Input

```
2
0 2
1 1 2
0 0.5 3

3
0 4
3 1 2
2 -1 0
1 1 2

0
```

Sample Output

```
2.41421356237
4.24264068712
```