A Amsterdam Distance

Your friend from Manhattan is visiting you in Amsterdam. Because she can only stay for a short while, she wants to see as many tourist attractions in Amsterdam in as little time as possible. To do that, she needs to be able to figure out how long it takes her to walk from one landmark to another. In her hometown, that is easy: to walk from point $m = (m_x, m_y)$ to point $n = (n_x, n_y)$ in Manhattan you need to walk a distance

$$|n_x - m_x| + |n_y - m_y|,$$

since Manhattan looks like a rectangular grid of city blocks. However, Amsterdam is not well approximated by a rectangular grid. Therefore, you have taken it upon yourself to figure out the shortest distances between attractions in Amsterdam. With its canals, Amsterdam looks much more like a half-disc, with streets radiating at regular angles from the center, and with canals running the arc of the circle at equally spaced intervals. A street corner is given by the intersection of a circular canal and a street which radiates from the city center.

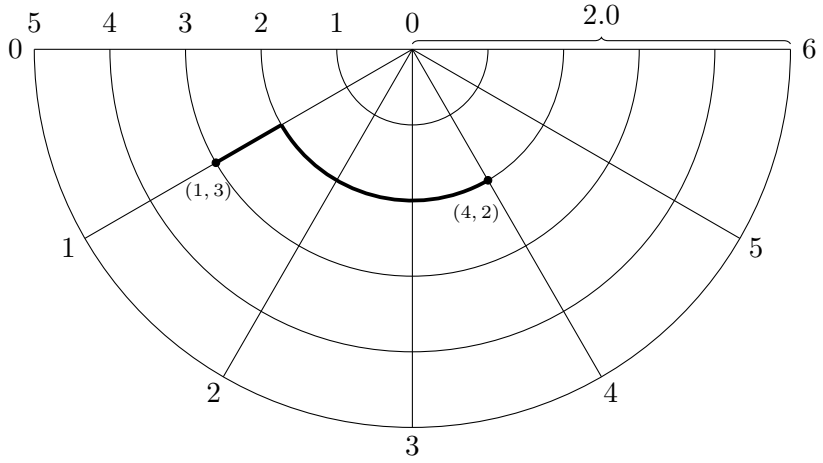


Figure 1: The first sample input.

Depending on how accurately you want to model the street plan of Amsterdam, you can split the city into more or fewer half rings, and into more or fewer segments. Also, to avoid conversion problems, you want your program to work with any unit, given as the radius of the half circle. Can you help your friend by writing a program which computes the distance between any two street corners in Amsterdam, for a particular approximation?

Input

The input consists of

- One line with two integers M, N and a floating point number R.
  - $1 \leq M \leq 100$ is the number of segments (or 'pie slices') the model of the city is split into.

  - $1 \leq N \leq 100$ is the number of half rings the model of the city is split into.
  - $1 \leq R \leq 1000$ is the radius of the city.
- One line with four integers, a_x, a_y, b_x, b_y, with $0 \leq a_x, b_x \leq M$, and $0 \leq a_y, b_y \leq N$, the coordinates of two corners in the model of Amsterdam.

Output

Output a single line containing a single floating point number, the least distance needed to travel from point a to point b following only the streets in the model. The result should have an absolute or relative error of at most 10^{-6}.

Sample Input 1	**Sample Output 1**
`6 5 2.0` `1 3 4 2`	`1.65663706143592`

Sample Input 2	**Sample Output 2**
`9 7 3.0` `1 5 9 5`	`4.28571428571429`

Sample Input 3	**Sample Output 3**
`10 10 1.0` `2 0 6 0`	`0`