A: Banjo

Having mastered the classic games Banjo-Kazooie and Banjo-Tooie, Bob has moved on to playing their secretly-released sequel, Banjo-Threeie. The game can be modelled with a 2-dimensional coordinate system. The player's objective is simply to walk from one point to another as the titular bear Banjo (carrying his bird pal, Kazooie, in his backpack), who can walk at a speed of 1 unit per second. Sounds pretty simple!

However, there is a twist. There is a lava pit in the way, which is perfectly circular. Though the pit is hot, Banjo can still walk across its surface at the same speed. However, if he spends more than ***t*** consecutive seconds in the pit, he'll lose all of his health points. If Banjo steps out of the pit for even a moment, however, his health points will reset, and he'll be able to step back into the pit if he wants to.

To make sure that he's playing the game as optimally as possible, Bob wants to know the minimum possible amount of time required to walk from the start to the end, without spending more than ***t*** consecutive seconds in the lava pit.

Input

There will be several test cases in the input. Each test case will consist of a single line with eight integers:

x1 y1 x2 y2 xc yc r t

where (***x1***,***y1***) is where Banjo starts, (***x2***,***y2***) is where Banjo must go, (***xc***,***yc***) is the center of the lava pit, ***r*** ($1 \le r \le 10{,}000$) is the radius of the pit, and ***t*** ($0 \le t \le 10{,}000$) is the number of consecutive seconds Banjo can spend in the pit before he loses all of his health points.

All coordinates will be in the range $-10{,}000 \le x, y \le 10{,}000$, and all three points in a test case will be unique. The starting and ending points will not be in the lava, nor will they be on the edge of the lava. The input will end with a line with eight 0s.

Output

For each test case, output a single number equal to the minimum time for Banjo to go from the start to the finish, given to exactly two decimal places, rounded. Output each number on its own line, with no spaces. Do not print any blank lines between outputs.

Sample Input	Sample Output
0 0 10 0 5 0 3 5 0 0 0 0 0 0 0 0	11.00