Ring

J3 / 150

Ivancho has plane with N points with integer coordinates and integer distance between each point and the point with coordinates (0,0) He wants to place two circles with center (0,0) and different non zero radiuses. He want the radiuses to be such points so that the area covered by the circle with the bigger radius is as small as possible and “between” the circles must be exactly K points. He doesn’t know how to find such radiuses so you have to write the program **ring**, that by given coordinates of the N points finds the minimal area of the figure so that between the two circles there are exactly K points and the area of this figure is minimal.

Note: The points „between“ the two circles can be placed on the border of the circle with the bigger radius but can’t be on the border of that with the smaller one. The both radiuses must be greater than 0. Each point is situated on integer distance of the point with coordinates (0,0).

Input

From the first row of the input file ring.in are entered 2 numbers – N and K – the count of the points and the number of points that must be situated „between“ the circles. On the next N rows there is pair of numbers representing the coordinates of each point.

Output

On the single row of the output file ring.out there must be only one number – the area of the figure Ivancho has to create divided by π. For example if the area of the figure is X*π you have to print X.

Constrains

1 <= N <= 100 000

-1 000 000, <= coordinates of each point <= 1 000 000

1 <= K <= 100 000

There are no points with equal coordinates.

There is no point with coordinates (0,0).

Example

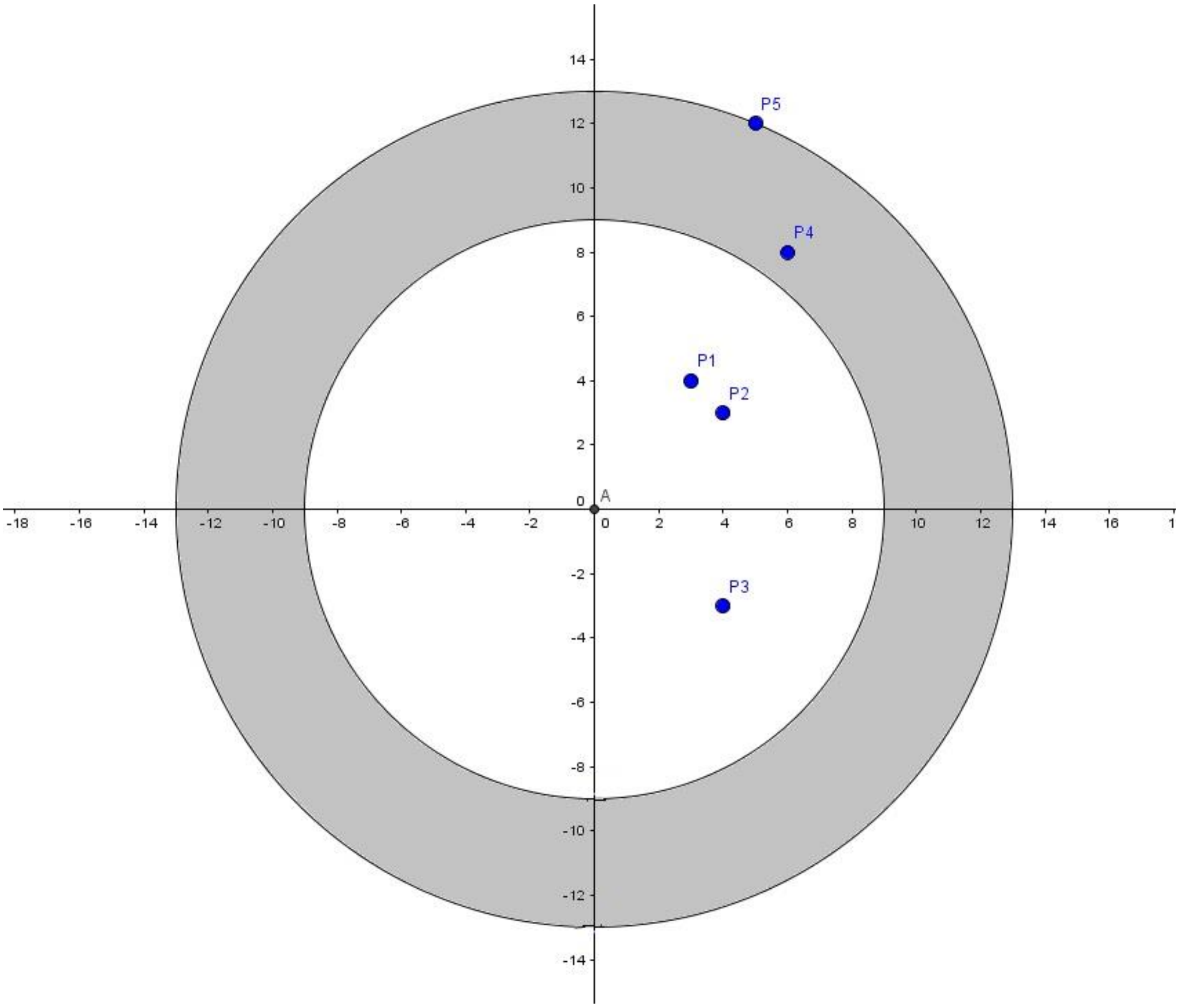
Input (ring.in)	**Output (ring.out)**

5 2
3 4
4 3
4 -3
6 8
5 12
	88

Explanation

Ring

J3 / 150



14
12
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12
-14
-18
-16
-14
-12
-10
-8
-6
-4
-2
0
2
4
6
8
10
12
14
16
1
A
P1
P2
P3
P4
P5