

k -means III

Task: Now the task is to solve a more realistic version of the last task. The points are now from \mathbb{R}^d with the Euclidean metric, and in addition, the task is dynamic: Instead of getting all centers upfront, we have to produce centers on the fly. More precisely, do the following: Read a stream of points. Compute a set of centers C ; in the beginning, C is empty. For each query point $x \in \mathbb{R}^d$, decide whether the closest point in C is at distance at most R from x . If so, ignore x . If not, add x to C .

Input: The first line contains two numbers. The first number $d \leq 100$ is an integer, the dimension of the points. The second number is a floating point number with three positions after the point, it is the threshold value R . After that, an (not previously specified) number of lines follows, each containing a point, given by d numbers separated by white spaces. The numbers are floating points with three positions after the point.

Output: Output a single integer: The number of points that are added to the center set.

Sample Input:

```
2 1.000
0.000 0.000
1.000 1.000
0.500 0.500
2.000 1.000
2.500 0.500
```

Sample Output:

```
3
```