**K Devices**

Max. Marks: 100

You are given the location of NN devices on a coordinate plane and an integer KK. Location of ithith device is donated by (Xi,Yi)(Xi,Yi). A modem is located at (0,0)(0,0). The range of modem is circular. All the devices within the range of the modem will connect to the modem. You have to find the minimum integral radius of the circular range of the modem such that at-least KK devices will connect to the modem.

**Input:**  
First line contains two integers, NN (1≤N≤105)(1≤N≤105) and KK (1≤K≤N)(1≤K≤N). Second line contains NN space separated integers denoting the array XX (−109≤Xi≤109)(−109≤Xi≤109). Third line contains NN space separated integers denoting the array YY (−109≤Yi≤109)(−109≤Yi≤109).

**Output:**  
Print one integer, denoting the minimum integral radius of the circular range of the modem such that at-least KK devices will connect to the modem.

**SAMPLE INPUT**

3 3

1 -1 1

1 -1 -1

**SAMPLE OUTPUT**

2

**Explanation**

All the devices are at distance √22 from the modem. So, the minimum integral radius of the circular range of the modem such that at-least 33 devices will connect to the modem will be 22.

**Time Limit:**1.0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded if any testcase passes.

**Allowed Languages:**C, C++, C++14, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Swift, Visual Basic

<https://www.hackerearth.com/challenge/competitive/august-circuits-17/algorithm/k-devices-96ab1c02/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

//int n = 3, k = 3;

//int[] x = { 1, -1, 1 };

//int[] y = { 1, -1, -1 };

string[] nk = Console.ReadLine().Trim().Split(' ');

int n = int.Parse(nk[0]);

int k = int.Parse(nk[1]);

long[] x = Array.ConvertAll(Console.ReadLine().Trim().Split(' '), e => long.Parse(e));

long[] y = Array.ConvertAll(Console.ReadLine().Trim().Split(' '), e => long.Parse(e));

List<long> distancias = new List<long>();

for (int i = 0; i < n; i++)

{

long dist = (long) Math.Ceiling( Math.Sqrt(x[i] \* x[i] + y[i] \* y[i]));

distancias.Add(dist);

//Console.WriteLine(dist);

}

distancias.Sort();

int cont = 0;

long min\_radius = -1;

for (int i = 0; cont<k && i < distancias.Count; i++)

{

//Console.WriteLine(distancias[i]);

min\_radius = distancias[i];

cont++;

}

Console.WriteLine(min\_radius);

Console.ReadLine();

}

}

}