
Parelles de punts consecutius que estan a prop**X67106_ca**

En aquest exercici us donem un programa que heu de completar. Al principi del programa es defineix un struct `Point`, que defineix un punt sobre el pla 2D amb coordenades enteres. El programa principal llegeix varis casos d'entrada. Cada cas consisteix en la descripció d'una llista de punts. Per a cada cas, el programa principal crida a una funció anomenada `compute` que és l'única que heu d'implementar (no heu de canviar res més). La funció `compute` rep la llista de punts com a paràmetre, i també rep un paràmetre enter `distance`, i retorna el nombre de parelles de punts que apareixen consecutivament a la llista, i tals que la distància (eucliciana) entre ells és menor o igual a `distance`.

Aquest és el codi que heu de completar:

```
#include <iostream>
#include <string>
#include <vector>

using namespace std;

struct Point {
    int x, y;
};

typedef vector<Point> VPoint;

// Afegiu funcions auxiliars si voleu.
// ...

// Pre:  v conté com a mínim un punt.
//       d >= 1
// Post: Retorna el nombre de parelles de punts que apareixen consecutivament a
//       i tals que la distància entre ells és menor o igual a d.
int compute(VPoint &v, int distance)
{
    // Implementeu aquesta funció.
    // ...
}

int main()
{
    int n, d;
    while (cin >> n >> d) {
        VPoint v(n);
        for (int i = 0; i < n; i++)
            cin >> v[i].x >> v[i].y;
        cout << compute(v, d) << endl;
    }
}
```

}

Entrada

L'entrada té varis casos. Cada cas comença amb dos naturals positius n , $distance$ majors o iguals a 1 en una primera línia. En una segona línia hi ha la descripció d'una llista d' n punts, amb les seves dues coordenades x,y (enteres). De fet, no us heu de preocupar massa per com son aquestes entrades perquè el programa que us donem ja té la part de lectura implementada. Només us heu de preocupar d'implementar la funció `compute` abans esmentada.

Sortida

Per a cada cas, el programa ha d'escriure en una nova línia el corresponent nombre de parelles de punts consecutius a la llista tals que la distància entre ells és menor o igual a $distance$. De fet, no us heu de preocupar massa per com son aquestes sortides perquè el programa que us donem ja té la part d'escriptura implementada. Només us heu de preocupar d'implementar la funció `compute` abans esmentada.

Exemple d'entrada 1

5 4	57 -46 69 -38 84 22 85 -10 -65 95 45 -12 18 28 -
10 6 2 1 4 0 6 3 1 8	9 66
5 1	24 -11 59 -48 36 -11 8 -16 24 68 -38 95 -38 2 8
5 3 7 4 9 10 2 0 10 8	4 66
5 20	51 -52 76 -44 93 -13 95 -92
0 4 6 0 10 3 10 10 7 10	17 66
10 5	24 -41 74 -52 48 32 -50 -67 71 58 -32 -6 25 -20
7 -1 -9 -3 6 6 5 8 2 4 10 0 10 -8	16 66
10 10	53 -52 -47 87 -82 61 -96 -38 -15 -16 51 -78 -35 2
-1 1 1 -9 0 -1 -4 1 2 7 -5 9 8 0 4	18 566 7 9 -3 -2
8 66	59 46 -1 20 50 62 55 -67 -38 77 -52 87 -30 -56 5
-56 -35 11 -5 78 41 -100 -44 -95 -37	18 66 -1 -7 7
17 66	-78 76 -99 50 75 88 70 -82 -4 -20 -60 -79 4 -74
-28 -18 21 30 -82 5 16 -34 -3 40 30	19 -38 7 -81 -73 -70
16 66	40 25 -4 -81 66 17 23 42 -82 74 -49 3 40 -6 76 -
59 27 59 75 -77 -52 -46 -8 33 -53 55	18 66 35 94 13 50 -46 -93 -23 96 58 -17 51 -79 91
20 66	71 1 -62 -27 91 -18 99 44 -93 -5 40 14 4 -44 19
-36 -50 -83 12 4 9 96 52 -87 -22 54	4 166 -48 62 10 -53 -31 94 98 -10 84 61 69 44 -49
5 66	-56 -60 -37 39 80 77 94 87
-21 -72 -35 -71 73 96 78 -93 -40 86	25 17 88 -100 95 31 48 -16 15 -42 -98 -92 9 -71 -47
10 66	-87 71 40 83 95 -71 56 -92 4 9 8 -17 8 60 90 -7
-82 -5 89 -29 68 35 -59 46 -95 -96 8	18 66
5 66	14 25 55 -28 -71 -86 80 -38 22 -11 -99 65 -30 -2
-42 83 86 -69 79 13 -62 89 -2 98	8 66 -24 86 -59 -50 -74 -9 77 -20 68
7 66	-30 -82 40 59 -80 5 -71 18 -14 27 -92 90
43 87 -72 10 21 -31 -45 -24 -76 93 52	18 66
7 66	32 -22 -75 -31 -23 -92 5 -67 59 11 -14 -42 30 -3
-25 28 -35 93 -16 -26 76 -31 5 4 32	13 91 6 85 -99
20 66	39 -60 44 100 26 -98 30 39 -35 -100 8 46 -9 45 6
-51 73 -74 -22 -68 47 -3 -12 73 21	19 66 -71
10 66	-47 60 -9 -62 -2 -19 32 -2 57 -16 -73 -55 -1 78
91 13 29 55 89 95 45 -7 -24 27 -20	8 66 31 -86 -24 -34 -61 54 32 -68 -63 -95 58 -44
8 66	-72 -12 46 -26 88 -27 77 83 -9 -26 -6 -100 -76 -5
100 -19 6 80 54 72 7 13 -90 5 -47 24	62 66 16 37 -9 45 -26 94 -53 92
6 66	-15 -10 -29 16
55 -96 -21 -65 -65 13 22 -23 -43 46	3 -67 -92 -38 78
20 66	-80 -100 -74 -39 -28 -3
	30 66 46
	-24 -85 11 15 11 35

7 66	Exemple de sortida 1																											
84 12 85 -25 36 -22 -75 11 27 -45 13	84	-10	53																									
3 66	2																											
-94 -94 -75 57 -67 -64	0																											
6 66	4																											
30 19 -45 -4 -70 20 57 15 72 41 77 56	2																											
20 66	6																											
13 -67 91 74 10 97 -14 -6 -64 89 -1	93	96	-26	49	-72	11	28	8	-71	83	54	-91	2	11	-25													
20 66	4																											
15 -18 7 89 82 35 -51 9 93 -71 -9 47	6	-62	43	-43	-37	16	-42	15	-56	75	79	-95	-69	-34														
15 66	3																											
-13 31 64 80 88 -74 -5 96 -9 39 20 -30	10	-5	1	-14	-28	86	-26	-34	-40	-25	74	-83	-69	-3														
12 66	2																											
-62 38 4 69 25 -19 -37 -30 72 -98 90	1	-59	-53	-60	-23	19	-74	2	35	-64	77	9	-47	58														
	1																											
	2																											
	2																											
	3																											
	2																											
	0																											
	7																											
	3																											
	2																											
	3																											
	2																											
	5																											
	6																											
	5																											
	3																											
	1																											
	2																											
	1																											
	3																											
	0																											
	3																											
	4																											
	7																											
	4																											
	3																											

Observació

Avaluació sobre 10 punts:

- Solució lenta: 5 punts.
- solució ràpida: 10 punts.

Entenem com a solució ràpida una que és correcta, de cost lineal i capaç de superar els jocs de proves públics i privats. Entenem com a solució lenta una que no és ràpida, però és correcta i capaç de superar els jocs de proves públics.

Informació del problema

Autoria: PRO1

Generació: 2026-01-25T17:38:52.605Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>