
Guillem Tell**U52362_ca**

En Guillem Tell entrena la seva punteria llençant fletxes contra una diana. El centre de la diana és el punt $(0,0)$ del pla, i cada fletxa acaba clavada en un punt (x,y) .

Escriu un programa que mantingui un *taulell* amb els k tirs més propers al centre de la diana. El programa rep dos enters, k i M . Després llegeix una seqüència de punts 2D (les posicions de cada fletxa). Cada M tirs, el programa ha de mostrar el taulell amb els k tirs més propers al centre, ordenats de més proper a més llunyà. Si dos tirs tenen la mateixa distància al centre, va primer el que té la coordenada x més petita, i si coincideixen, el que té la y més petita.

Si en algun moment hi ha menys de k tirs, el taulell mostra només els que hi hagi.

Si al final de l'entrada la quantitat de tirs no és múltiple de M , cal mostrar el taulell una última vegada.

Entrada

La primera línia conté dos enters k i M . Les línies següents contenen dos reals x i y cadascuna, les coordenades de cada tir.

Sortida

Cada M tirs, el taulell amb els k tirs més propers al centre (o menys si no n'hi ha prou), un per línia amb el format (x, y) . Entre dues taules consecutives, una línia amb ---.

Observació

El centre d'interès en aquest problema és l'**eficiència**.

No cal calcular arrels quadrades per comparar distàncies al centre: $\sqrt{x^2 + y^2} < \sqrt{x'^2 + y'^2}$ és equivalent a $x^2 + y^2 < x'^2 + y'^2$.

Si feu servir un *Heap*, penseu quin signe ha de tenir la prioritat perquè el tir "pitjor" dels k millors estigui sempre accessible al capdamunt.

Als fitxers públics (icona del gatet) trobaràs els contenidors de PRO2 (`stack.hh`, `queue.hh`, `heap.hh`, i la seva dependència `assert.hh`), un `program.cc` buit per començar, un `Makefile` i el directori `.vscode` amb la configuració per compilar i debuggar amb VSCode.

Implementa el programa al fitxer `program.cc` i envia'l al Jutge. Pots fer servir qualsevol contenidor de la STL i els de PRO2 que necessitis.

Exemple d'entrada 1

```
3 4
1.0 2.0
-0.5 0.3
3.0 4.0
0.1 -0.1
```

Exemple de sortida 1

```
(0.1, -0.1)
(-0.5, 0.3)
(1, 2)
```

Exemple d'entrada 2

```
5 3
1.0 1.0
-2.0 0.5
0.1 0.2
0.0 0.5
-1.0 -1.05
3.0 3.0
-0.1 0.1
```

Exemple de sortida 2

```
(0.1, 0.2)
(1, 1)
(-2, 0.5)
---
(0.1, 0.2)
(0, 0.5)
(1, 1)
(-1, -1.05)
(-2, 0.5)
---
(-0.1, 0.1)
(0.1, 0.2)
(0, 0.5)
(1, 1)
(-1, -1.05)
```

Informació del problema

Autoria: Pau Fernández

Generació: 2026-04-06T16:55:26.717Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>