

---

**Distàncies (II)****X96957\_ca**

---

Tenim un mercat ambulant i volem distribuir el nostre producte a  $n$  pobles utilitzant una aproximació de l'algorisme del veí més proper. Per fer-ho, disposem d'una graella bidimensional de  $100 \times 100$  unitats de longitud, entre les quals hi ha distribuïdes  $n+1$  coordenades que es troben emmagatzemades en una llista de valors enters com la següent:

$$coord = [x_{ref}, y_{ref}, x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, \dots, x_n, y_n]$$

Volem crear una llista `distancies` que guardi totes les distàncies (valors reals amb 2 decimals) des de la coordenada de referència `x_ref, y_ref` (que es troben a la primera i segona posició de la llista respectivament) fins a la resta de coordenades dels diferents pobles, ordenades de menor a major distància.

Per calcular la distància euclidiana entre dues coordenades

$$(x_1, y_1) \text{ i } (x_2, y_2)$$

la distància entre elles ( $d$ , valor real) es pot calcular de la següent manera:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

NOTA: Per arrodonir un número real a un número de decimals específic, podem utilitzar la funció: `round(num_real, número_de_decimals)`.

Per exemple, per fer l'arrodoniment a 2 decimals, si `valor = 4.35342135`, llavors `round(valor, 2)` retornarà `4.35`.

La pràctica consisteix en dos exercicis (Part 1 i Part2) que son independents

**PART 2**

Donada una llista de distancies de valors reals arrodonits a 2 decimals ordenada de menor a major, creada a la Part1, cal crear ara la funció `dades_distancia`, que haurà de retornar el següents valors reals, arrodonits a 2 decimals, en relació amb totes les distàncies entre la coordenada de referència i els  $n$  pobles:

1. La distancia mínima,
2. La distancia màxima,
3. La distancia mitjana,
4. Les unitats de longitud totals que recorrem si tenim en compte que per cada poble que visitem, cal fer l'anada i tornada des de la coordenada de referència.

**Observació**

Només cal que enviïs el fitxer `solution.py` amb la funció `dades_distancia` (i les funcions auxiliars que hagi fet) que et demanem i prou. El fitxer `main.py` et pot servir per a fer la teva solució, però no cal que n'enviïs el contingut.

## Entrada

llista de distàncies (valors reals amb 2 decimals) de la forma:

$$[d_1, d_2, d_3, d_4, \dots, d_n]$$

## Sortida

Quatre valors reals, el la mateixa línia, arrodonits a 2 decimals, amb els valors:

1. Distància mínima,
2. Distància màxima,
3. Distància mitjana,
4. Unitats de longitud totals que recorrem si tenim en compte que, per cada poble que visitem, cal fer l'anada i tornada des de la coordenada de referència.

### Exemple d'entrada 1

28.28 42.43 98.99 113.14 141.42

### Exemple d'entrada 2

1.41 58.19 67.54 72.01

### Exemple de sortida 1

[28.28, 42.43, 98.99, 113.14, 141.42]  
28.28 141.42 84.85 848.52

### Exemple de sortida 2

[1.41, 58.19, 67.54, 72.01]  
1.41 72.01 49.79 398.3

## Informació del problema

Autoria: Jose Antonio Roman

Generació: 2026-01-25T20:02:13.696Z

© Jutge.org, 2006–2026.

<https://jutge.org>