

On souhaite programmer une fonction donnant la distance la plus courte entre un point de départ et une liste de points. Les points sont tous à coordonnées entières.

Les points sont donnés sous la forme d'un tuple de deux entiers.

La liste des points à traiter est donc un tableau, non vide, de tuples.

On rappelle que la distance entre deux points du plan de coordonnées  $(x ; y)$  et  $(x' ; y')$  est donnée par la formule :

$$d = \sqrt{(x - x')^2 + (y - y')^2}.$$

On importe pour cela la fonction racine carrée (`sqrt`) du module `math` de Python.

Compléter le code des fonctions `distance` et `plus_courte_distance` fournies à la page suivante pour qu'elles répondent à leurs spécifications.

```
from math import sqrt    # import de la fonction racine carrée

def distance(point1, point2):
    """ Calcule et renvoie la distance entre deux points. """
    return sqrt((...) ** 2 + (...) ** 2)

def plus_courte_distance(tab, depart):
    """ Renvoie le point du tableau tab se trouvant à la plus
    courte distance du point depart. """
    point = tab[0]
    min_dist = ...
    for i in range (1, ...):
        if distance(tab[i], depart)...:
            point = ...
            min_dist = ...
    return point
```

**Exemples :**

```
>>> distance((1, 0), (5, 3))
```

```
5.0
```

```
>>> distance((1, 0), (0, 1))
```

```
1.4142135623730951
```

```
>>> plus_courte_distance([(7, 9), (2, 5), (5, 2)], (0, 0))  
(2, 5)
```

```
>>> plus_courte_distance([(7, 9), (2, 5), (5, 2)], (5, 2))  
(5, 2)
```