

```

from math import sqrt    # import de la fonction racine carrée

def distance(point1, point2):
    """ Calcule et renvoie la distance entre deux points. """
    assert isinstance(point1, tuple)
    assert len(point1) == 2
    assert isinstance(point1[0], int)
    assert isinstance(point1[1], int)
    assert isinstance(point2, tuple)
    assert len(point2) == 2
    assert isinstance(point2[0], int)
    assert isinstance(point2[1], int)
    return sqrt((point1[0]-point2[0])**2 + (point1[1]-point2[1])**2)

assert distance((1, 0), (5, 3)) == 5.0, "erreur de calcul"

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 = distance(point, depart)
    for i in range(1, len(tab)):
        if distance(tab[i], depart) < min_dist:
            point = tab[i]
            min_dist = distance(tab[i], depart)
    return point

assert plus_courte_distance([(7, 9), (2, 5), (5, 2)], (0, 0)) == (2, 5), "erreur"

import unittest
from random import randint

class validation(unittest.TestCase):
    def test_distance_assert_tuple1(self):
        a, b, c, d = [randint(-100, 100) for _ in range(4)]
        self.assertRaises( AssertionError, distance, [a, b], (c, d) )
        self.assertRaises( AssertionError, distance, (a, b), [c, d] )

    def test_distance_assert_couple(self):
        a, b, c, d, e = [randint(-100, 100) for _ in range(5)]
        self.assertRaises(AssertionError, distance, (a, b, c), (d, e))
        self.assertRaises(AssertionError, distance, (a, b), (c, d, e))

    def test_distance_assert_int(self):
        a = chr(randint(65, 90))
        n1, n2, n3 = [randint(-100, 100) for _ in range(3)]
        self.assertRaises(AssertionError, distance, (a, n1), (n2, n3))
        self.assertRaises(AssertionError, distance, (n1, a), (n2, n3))
        self.assertRaises(AssertionError, distance, (n1, n2), (a, n3))
        self.assertRaises(AssertionError, distance, (n1, n2), (n3, a))

if __name__ == '__main__':
    unittest.main()

```