

```
liste_eleves = ['a','b','c','d','e','f','g','h','i','j']
liste_notes = [1, 40, 80, 60, 58, 80, 75, 80, 60, 24]

def meilleures_notes():
    note_maxi = 0
    nb_eleves_note_maxi = 0
    liste_maxi = []

    for compteur in range(len(liste_eleves)):
        if liste_notes[compteur] == note_maxi:
            nb_eleves_note_maxi = nb_eleves_note_maxi + 1
            liste_maxi.append(liste_eleves[compteur])
        if liste_notes[compteur] > note_maxi:
            note_maxi = liste_notes[compteur]
            nb_eleves_note_maxi = 1
            liste_maxi = [liste_eleves[compteur]]

    return (note_maxi,nb_eleves_note_maxi,liste_maxi)

# version pour codeboard.IOErrorimport Root.src.main as mn
# import unittest

# class validation(unittest.TestCase):
#     def test_1_statique(self):
#         print(mn.meilleures_notes())
#         self.assertTupleEqual(mn.meilleures_notes(), (80, 3, ['c', 'f', 'h']))

#     def test_2_alea(self):
#         from random import randint
#         n = randint(5, 20)
#         mn.liste_notes = [randint(0, 2*n//3) for _ in range(n)]
#         mn.liste_eleves = []
#         while len(mn.liste_eleves) != n:
#             nom = chr(randint(97, 122))
#             if nom not in mn.liste_eleves:
#                 mn.liste_eleves.append(nom)

#         def meilleures_notes_sol():
#             note_maxi = 0
#             nb_eleves_note_maxi = 0
#             liste_maxi = []

#             for compteur in range(len(mn.liste_eleves)):
#                 if mn.liste_notes[compteur] == note_maxi:
#                     nb_eleves_note_maxi = nb_eleves_note_maxi + 1
#                     liste_maxi.append(mn.liste_eleves[compteur])
#                 if mn.liste_notes[compteur] > note_maxi:
#                     note_maxi = mn.liste_notes[compteur]
#                     nb_eleves_note_maxi = 1
#                     liste_maxi = [mn.liste_eleves[compteur]]

#             return (note_maxi,nb_eleves_note_maxi,liste_maxi)

#         self.assertTupleEqual(mn.meilleures_notes(), meilleures_notes_sol())

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