

**Exercice 1 :** L'assertion évite ici de faire une division par zéro.

```
1 def moyenne(notes: list) -> int:
2     """
3     calcule la moyenne pondérée des notes
4
5     Args:
6         notes (list): liste de tuples (note, coef)
7
8     Returns:
9         int: moyenne
10    """
11    assert len(notes) > 0, "Il faut au moins une note."
12    somme = 0
13    coefficients = 0
14    for couple in notes:
15        somme += couple[0]*couple[1]
16        coefficients += couple[1]
17    print(somme, coefficients)
18    return somme/coefficients
```

**Exercice 2 :**

```
1 def recherche_minmax(tab: list)->dict:
2     """
3     renvoie les valeurs min et max
4     de tab, sous forme d'un dictionnaire
5     """
6     if not tab:
7         return {"min": None, "max": None}
8     else:
9         dico = {"min": tab[0], "max": tab[0]}
10        for val in tab:
11            if val < dico["min"]:
12                dico["min"] = val
13            if val > dico["max"]:
14                dico["max"] = val
15        return dico
```

**Exercice 3 :**

```
1 def moyenne(nom):
2     if nom in resultats:
3         notes = resultats[nom]
4         total_points = 0
5         total_coefficients = 0
6         for valeurs in notes.values():
7             note , coefficient = valeurs
8             total_points = total_points + note * coefficient
9             total_coefficients = total_coefficients + coefficient
10        return round( total_points / total_coefficients , 1 )
11    else:
12        return -1
```