
Tests boîte blanche

Corrigés

1. Exercice : tests « boîte blanche »

Contexte :

Un programme simple est écrit pour calculer la moyenne d'une liste de notes d'étudiants et vérifier si un étudiant est reçu ou non. Si la moyenne est supérieure ou égale à 10, l'étudiant est reçu ; sinon, il est recalé.

Voici le code du programme :

```
def calcul_moyenne(notes):  
    somme = 0  
    for note in notes:  
        somme += note  
    moyenne = somme / len(notes)  
    return moyenne  
  
def est_recu(notes):  
    moyenne = calcul_moyenne(notes)  
    if moyenne >= 10:  
        return True  
    else:  
        return False
```

Consignes :

1. Analyse du Code :

- Lisez attentivement le code et identifiez les différentes conditions et cas particuliers qui pourraient affecter le comportement du programme.

2. Identification des Cas de Test :

- Réfléchissez aux différentes situations qui peuvent survenir lors de l'exécution de ce programme. Considérez les cas particuliers comme une liste vide ou des notes limites (par exemple, une moyenne exactement égale à 10).

3. Création des Jeux de Tests :

- Proposez une série de jeux de tests (entrées et résultats attendus) pour vérifier le bon fonctionnement du programme. Chaque jeu de tests doit permettre de tester un aspect différent du programme, en particulier les conditions identifiées précédemment.

4. Réalisation des Tests Unitaires :

- Écrivez des tests unitaires pour vérifier les différentes parties du programme. Utilisez le framework pytest pour automatiser ces tests.

Pour aider à tester une moyenne :

```
pytest.approx(6.33, 0.01)
```

Convention de nommage avec pytest :

```
def test_xxxxxx():
```

Indications :

- Pensez à des cas où la liste de notes est vide, où elle contient des notes égales ou supérieures à 10, et où elle ne contient que des notes inférieures à 10.
- Vérifiez comment le programme se comporte avec différents jeux de notes et assurez-vous qu'il retourne les résultats attendus.

```
import pytest
from notes import calcul_moyenne, est_recu # On suppose que le code précédent est
dans un fichier `notes.py`

# Test de la fonction calcul_moyenne


# Test de la fonction est_recu


if __name__ == "__main__":
    pytest.main()
```

Corrigé

Exemple de Tests à Proposer :

Voici quelques idées de cas de test que vous pourriez envisager :

1. **Liste vide** : Vérifiez le comportement du programme lorsque la liste des notes est vide.
2. **Moyenne inférieure à 10** : Une liste de notes dont la moyenne est strictement inférieure à 10.
3. **Moyenne égale à 10** : Une liste de notes dont la moyenne est exactement égale à 10.
4. **Moyenne supérieure à 10** : Une liste de notes dont la moyenne est strictement supérieure à 10.

```

import pytest
from notes import calcul_moyenne, est_recu # On suppose que le code précédent est dans
un fichier `notes.py`

# Test de la fonction calcul_moyenne

def test_moyenne_vide():
    # Cas 1 : Liste vide
    assert calcul_moyenne([]) == 0

def test_moyenne_inf_10():
    # Cas 2 : Moyenne strictement inférieure à 10
    assert calcul_moyenne([5, 6, 8]) == pytest.approx(6.33, 0.01)

def test_moyenne_egal_10():
    # Cas 3 : Moyenne exactement égale à 10
    assert calcul_moyenne([10, 10, 10]) == 10

def test_moyenne_sup_10():
    # Cas 4 : Moyenne supérieure à 10
    assert calcul_moyenne([12, 14, 16]) == pytest.approx(14, 0.01)

# Test de la fonction est_recu

def test_recu_vide():
    # Cas 1 : Liste vide (étudiant recalé, car moyenne = 0)
    assert est_recu([]) == False

def test_recu_moyenne_inf_10():
    # Cas 2 : Moyenne strictement inférieure à 10 (étudiant recalé)
    assert est_recu([5, 6, 8]) == False

def test_recu_moyenne_egal_10():
    # Cas 3 : Moyenne exactement égale à 10 (étudiant reçu)
    assert est_recu([10, 10, 10]) == True

def test_recu_moyenne_sup_10():
    # Cas 4 : Moyenne supérieure à 10 (étudiant reçu)
    assert est_recu([12, 14, 16]) == True

if __name__ == "__main__":
    pytest.main()

```

Explication des tests :

1. Tests de la fonction calcul_moyenne :

- **test_moyenne_vide()** : Vérifie que la fonction retourne 0 si la liste est vide.
- **test_moyenne_inf_10()** : Teste une liste où la moyenne est inférieure à 10.
- **test_moyenne_egal_10()** : Vérifie que la fonction retourne exactement 10 si la moyenne des notes est 10.
- **test_moyenne_sup_10()** : Vérifie une liste où la moyenne est supérieure à 10.

2. Tests de la fonction est_recu :

- **test_recu_vide()** : Vérifie que l'étudiant est recalé si la liste est vide (moyenne = 0).
- **test_recu_moyenne_inf_10()** : Vérifie que l'étudiant est recalé si la moyenne est inférieure à 10.
- **test_recu_moyenne_egal_10()** : Vérifie que l'étudiant est reçu si la moyenne est exactement égale à 10.
- **test_recu_moyenne_sup_10()** : Vérifie que l'étudiant est reçu si la moyenne est supérieure à 10.