

Examen de Test Unitaire en Python

Question 1:

```
PS C:\Users\DÉVELOPPEUR\sacha_django_project> python -m unittest test_est_premier.py
.....
-----
Ran 14 tests in 0.002s

OK
```

Ces tests couvrent différents cas :

- Nombre négatif
- Nombre zéro
- Nombre un
- Nombre petit non premier (ex: 4)
- Nombre petit premier (ex: 2)
- Nombre moyen non premier (ex: 10)
- Nombre moyen premier (ex: 13)
- Nombre grand non premier (ex: 100)
- Nombre grand premier (ex: 97)
- Nombre très grand non premier (ex: 1000)
- Nombre très grand premier (ex: 997)
- Nombre premier à trois chiffres
- Nombre non premier à trois chiffres
- Nombre premier négatif

Question 2:

```
PS C:\Users\DÉVELOPPEUR\sacha_django_project> python -m unittest test_compter_mots.py
.....
-----
Ran 9 tests in 0.002s

OK
```

Ces tests couvrent différents cas :

- Phrase vide
- Phrase avec espaces supplémentaires
- Phrase normale
- Phrase avec des caractères spéciaux seulement
- Phrase avec des nombres
- Phrase avec des caractères spéciaux et des mots
- Phrase avec des mots accentués
- Mots avec des apostrophes
- Mots avec des tirets

Question 3:

```
PS C:\Users\DÉVELOPPEUR\sacha_django_project> python -m unittest test_compte_bancaire.py
....
-----
Ran 4 tests in 0.001s

OK
```

Ces tests couvrent différents cas :

- Dépôt d'une somme sur un compte existant.
- Dépôt d'une somme nulle sur un compte existant.
- Retrait d'une somme d'un compte avec un solde suffisant.
- Retrait d'une somme nulle d'un compte existant.
- Tentative de retrait d'une somme supérieure au solde du compte.
- Obtention du solde d'un compte non vide.
- Obtention du solde d'un compte vide.

Question 4:

```
PS C:\Users\DÉVELOPPEUR\sacha_django_project> python -m unittest test_somme_liste.py
.....
-----
Ran 9 tests in 0.002s

OK
```

Ces tests couvrent différents cas :

- Une liste vide.
- Une liste avec des éléments positifs.
- Une liste avec des éléments négatifs.
- Une liste avec des éléments positifs et négatifs mélangés.
- Une liste avec un seul élément.
- Une liste avec des nombres décimaux.
- Une liste avec des éléments non numériques.
- Une liste avec des éléments de type None.
- Une liste avec des éléments de types numériques différents.

Question 5:

```
PS C:\Users\DÉVELOPPEUR\sacha_django_project> python -m unittest test_rectangle.py
.....
-----
Ran 8 tests in 0.002s

OK
```

Ces tests couvrent différents cas :

- Calcul du périmètre pour un rectangle donné.
- Calcul de la surface pour un rectangle donné.
- Calcul du périmètre pour un rectangle avec des dimensions nulles.
- Calcul de la surface pour un rectangle avec des dimensions nulles.
- Calcul du périmètre pour un rectangle avec des dimensions égales.
- Calcul de la surface pour un rectangle avec des dimensions égales.
- Calcul du périmètre pour un rectangle avec des dimensions décimales.
- Calcul de la surface pour un rectangle avec des dimensions décimales.

Question 6:

```
PS C:\Users\DÉVELOPPEUR\sacha_django_project> python -m unittest test_est_palindrome
.....
-----
Ran 5 tests in 0.001s

OK
```

Ces tests couvrent plusieurs cas :

- Un palindrome normal sans espace.
- Un palindrome avec des espaces.
- Une chaîne qui n'est pas un palindrome.
- Un palindrome avec des caractères sensibles à la casse.
- Une chaîne vide.

```
21 def est_palindrome(chaine):
22     chaine = chaine.lower() #Pour gérer la casse
23     chaine_inverse = chaine[::-1]
24     return chaine == chaine_inverse
```

Je me suis permis de modifier la fonction `est_palindrome()` afin de gérer la casse.

Question 7:

```
PS C:\Users\DÉVELOPPEUR\sacha_django_project> python -m unittest test_calculer_moyenne
.....
-----
Ran 5 tests in 0.001s

OK
```

Ces tests couvrent différents scénarios :

Une liste non vide avec des nombres positifs, négatifs et un mélange.

Une liste vide.

Une liste avec un seul élément.

Une liste avec tous les éléments égaux à zéro.

Une liste avec des nombres décimaux.