

TP Tests Unitaires

Question 1

Tests si un nombre est premier ou pas dans le cas de grands nombres, nombres premiers ou non, nombres négatifs et 0 et 1.

```
MacBook-Pro-de-Hugo:Tests_Unitaires hugo$ python3 -m unittest -v tests/test_est_premier.py
test_grands_nombres (tests.test_est_premier.TestEstPremier.test_grands_nombres)
Vérifie si des grands nombres retournent correctement True ou False ... ok
test_nombres_negatifs (tests.test_est_premier.TestEstPremier.test_nombres_negatifs)
Vérifie si des nombres négatifs retournent correctement False ... ok
test_nombres_non_premiers (tests.test_est_premier.TestEstPremier.test_nombres_non_premiers)
Vérifie si des nombres non premiers retournent correctement False ... ok
test_nombres_premiers (tests.test_est_premier.TestEstPremier.test_nombres_premiers)
Vérifie si des nombres premiers retournent correctement True ... ok
test_zero_et_un (tests.test_est_premier.TestEstPremier.test_zero_et_un)
Vérifie si 0 et 1 retournent correctement False ... ok

-----
Ran 5 tests in 0.000s

OK
```

Question 2

Tests pour compter le nombre de mots lorsque c'est vide, qu'il y a des caractères spéciaux, Un seul mot, des espaces en plus et de la ponctuation.

```
MacBook-Pro-de-Hugo:Tests_Unitaires hugo$ python3 -m unittest -v tests/test_compter_mots.py
test_mots_espaces_en_plus (tests.test_compter_mots.TestCompterMots.test_mots_espaces_en_plus)
Vérifie si une phrase avec des espaces en plus retourne le nombre correct de mots. ... ok
test_phrase_caracteres_speciaux (tests.test_compter_mots.TestCompterMots.test_phrase_caracteres_speciaux)
Vérifie si une phrase avec des caractères spéciaux retourne le nombre correct de mots ... ok
test_phrase_ponctuation (tests.test_compter_mots.TestCompterMots.test_phrase_ponctuation)
Vérifie si une phrase avec de la ponctuation retourne le nombre correct de mots ... ok
test_phrase_vide (tests.test_compter_mots.TestCompterMots.test_phrase_vide)
Vérifie si une phrase vide retourne 0 ... ok
test_un_mot (tests.test_compter_mots.TestCompterMots.test_un_mot)
Vérifie si une phrase avec un seul mot retourne 1 ... ok

-----
Ran 5 tests in 0.000s

OK
```

Question 3

Tests sur le compte bancaire en ajoutant de l'argent, en retirant de l'argent avec un solde suffisant et insuffisant et pour obtenir le solde.

```
MacBook-Pro-de-Hugo:Tests_Unitaires hugo$ python3 -m unittest -v tests/test_compte_bancaire.py
test_deposer (tests.test_compte_bancaire.TestCompteBancaire.test_deposer)
Vérifie si la méthode deposer fonctionne correctement ... ok
test_obtenir_solde (tests.test_compte_bancaire.TestCompteBancaire.test_obtenir_solde)
Vérifie si la méthode obtenir_solde retourne correctement le solde du compte ... ok
test_retirer_solde_insuffisant (tests.test_compte_bancaire.TestCompteBancaire.test_retirer_solde_insuffisant)
Vérifie si la méthode retirer génère une erreur en cas de solde insuffisant ... ok
test_retirer_solde_suffisant (tests.test_compte_bancaire.TestCompteBancaire.test_retirer_solde_suffisant)
Vérifie si la méthode retirer fonctionne correctement avec un solde suffisant ... ok

-----
Ran 4 tests in 0.000s

OK
```

Question 4

Tests sur la somme d'une liste avec une addition de nombres positifs, négatifs, mixtes, liste vide et lorsqu'il y a un seul élément dans la liste.

```
MacBook-Pro-de-Hugo:Tests_Unitaires hugo$ python3 -m unittest -v tests/test_somme_liste.py
test_liste_mixte (tests.test_somme_liste.TestSommeListe.test_liste_mixte)
Vérifie si la fonction retourne la somme correcte pour une liste de nombres mixtes (positifs et négatifs) ... ok
test_liste_negatifs (tests.test_somme_liste.TestSommeListe.test_liste_negatifs)
Vérifie si la fonction retourne la somme correcte pour une liste de nombres négatifs ... ok
test_liste_positifs (tests.test_somme_liste.TestSommeListe.test_liste_positifs)
Vérifie si la fonction retourne la somme correcte pour une liste de nombres positifs ... ok
test_liste_un_seul_element (tests.test_somme_liste.TestSommeListe.test_liste_un_seul_element)
Vérifie si la fonction retourne le seul élément de la liste lorsqu'il y a un seul élément dans la liste ... ok
test_liste_vide (tests.test_somme_liste.TestSommeListe.test_liste_vide)
Vérifie si la fonction retourne 0 pour une liste vide ... ok

-----
Ran 5 tests in 0.000s

OK
```

Question 5

Tests sur le périmètre et la surface d'un rectangle.

```
MacBook-Pro-de-Hugo:Tests_Unitaires hugo$ python3 -m unittest -v tests/test_rectangle.py
test_calculer_perimetre (tests.test_rectangle.TestRectangle.test_calculer_perimetre)
Vérifie si la méthode calculer_perimetre() calcule correctement le périmètre du rectangle ... ok
test_calculer_surface (tests.test_rectangle.TestRectangle.test_calculer_surface)
Vérifie si la méthode calculer_surface() calcule correctement la surface du rectangle ... ok

-----
Ran 2 tests in 0.000s

OK
```

Question 6

J'ai modifié la fonction car le cas de la casse n'était pas traité, j'ai donc traité le cas des majuscules/minuscules, la suppression des espaces et de la ponctuation

Tests lorsque le mot est un palindrome ou non, un palindrome avec espaces et lorsque le palindrome contient des majuscules et minuscules.

```
MacBook-Pro-de-Hugo:Tests_Unitaires hugo$ python3 -m unittest -v tests/test_est_palindrome.py
test_non_palindrome (tests.test_est_palindrome.TestEstPalindrome.test_non_palindrome)
Vérifie si une chaîne de caractères qui n'est pas un palindrome est détectée correctement ... ok
test_palindrome_avec_espaces (tests.test_est_palindrome.TestEstPalindrome.test_palindrome_avec_espaces)
Vérifie si une chaîne de caractères avec des espaces est détectée correctement comme un palindrome ... ok
test_palindrome_majuscules_minuscules (tests.test_est_palindrome.TestEstPalindrome.test_palindrome_majuscules_minuscules)
Vérifie si la casse des lettres est ignorée lors de la détection des palindromes ... ok
test_palindrome_normal (tests.test_est_palindrome.TestEstPalindrome.test_palindrome_normal)
Vérifie si une chaîne de caractères palindrome est détectée correctement ... ok

-----
Ran 4 tests in 0.000s

OK
```

Question 7

Tests sur le calcul d'une moyenne de nombres, sur une liste vide, sur un seul élément, avec nombres décimaux et nombres négatifs.

```
MacBook-Pro-de-Hugo:Tests_Unitaires hugo$ python3 -m unittest -v tests/test_calcul_moyenne.py
test_moyenne (tests.test_calcul_moyenne.TestCalculerMoyenne.test_moyenne)
Vérifie si la moyenne est correctement calculée pour une liste non vide ... ok
test_moyenne_decimaux (tests.test_calcul_moyenne.TestCalculerMoyenne.test_moyenne_decimaux)
Vérifie si la moyenne est correctement calculée pour une liste de nombres décimaux ... ok
test_moyenne_liste_vide (tests.test_calcul_moyenne.TestCalculerMoyenne.test_moyenne_liste_vide)
Vérifie si une erreur ValueError est levée lorsque la liste est vide ... ok
test_moyenne_negatifs (tests.test_calcul_moyenne.TestCalculerMoyenne.test_moyenne_negatifs)
Vérifie si la moyenne est correctement calculée pour une liste de nombres négatifs ... ok
test_moyenne_un_seul_element (tests.test_calcul_moyenne.TestCalculerMoyenne.test_moyenne_un_seul_element)
Vérifie si la moyenne est correctement calculée pour une liste avec un seul élément ... ok

-----
Ran 5 tests in 0.000s

OK
```