TP 7 - Tests unitaires

L'objectif de ce TP est d'utiliser et de créer des tests unitaires avec le *framework pytest* pour les modules que vous avez développés jusqu'à présent.

Vous lancerez vos tests unitaires depuis la racine de votre projet (à vous de bien configurer le `PYTHONPATH`).

1 Tests unitaires pour les modules `compteur` et `file_de_priorite`

Déposez les tests unitaires disponibles sur moodle dans le répertoire `tests`. Lancez pytest et vérifiez que tous les tests passent.

2 Tests unitaires pour le module `code_binaire`

Créez le fichier `test_code_binaire.py` dans le répertoire `tests`.

Vous ajouterez successivement à votre script les tests suivants (vous ne passerez à l'étape suivante $n+1$ que si votre code passe tous les tests des étapes 1 à n) :

1. testez le comportement de l'opérateur d'égalité (==) avec des codes binaires créés à partir d'un bit puis de plusieurs ;
2. testez que l'ajout d'un bit se fait bien à la fin ;
3. testez le comportement de l'opérateur `[ ]` en lecture, en testant les cas ou l'indice est un `int` ou un `slice` ;
4. testez le comportement de l'opérateur `[ ]` en écriture, en testant les cas où l'indice est un `int` ou un `slice`, les cas où on affecte une séquence de bits ou un code bianire, et les cas ou le changement réduit ou agrandit la longueur du code binaire (tout en respectant la documentation) ;
5. testez le comportement de l'instruction `del` en testant les cas où l'indice est un `int` ou un `slice` ;
6. testez le comportement de l'opérateur +, en vérifiant bien que l'exception `TypeError` est levée lorsque l'opérande de droite n'est pas un code binaire ;
7. testez le comportement de la fonction `len` avec des codes binaires créés à partir de bits, par ajout de bits ou par concaténation de code binaire ;
8. testez le fait qu'il y a l'exception `AuMoinsUnBitErreur` qui est levée lorsque l'on enlève trop de bits (il en faut toujours au moins un) via l'utilisation de la fonction `del` ou de l'opérateur `[ ]` en écriture ;
9. testez la représentation formelle qui doit permettre de retrouver le même (au sens de l'égalité) code binaire si on évalue (fonction `eval`) la chaîne de caractères obtenue ;
10. testez la représentation informelle.