TP #3

LE VIF DU SUJET

Listes et modules.

Création d'un module de fonctions sur les ensembles.

Objectif du TP : Écrire un ensemble de fonction permettant la création et la manipulation des ensembles mathématiques en se basant sur l'utilisation des listes de Python.

Selon Wikipedia, un **ensemble** désigne intuitivement une collection d'objets (les **éléments** de l'ensemble), « une multitude qui peut être comprise comme un tout ».

Plus précisément, un ensemble est une collection d'objets **uniques**, c'est-à-dire que l'ajout d'un élément à un ensemble contenant déjà un tel élément ne modifiera pas l'ensemble.

Comme nous souhaitons développer un module de gestion des ensembles, nous allons avoir besoin d'un certain nombre de fonctions permettant de les manipuler. En voici la liste :

- Création d'un ensemble;
- Appartenance d'un élément à un ensemble;
- Intersection de deux ensembles;
- Union de deux ensembles;
- Différence de deux ensembles;
- Différence symétrique de deux ensembles;
- Retrait d'un élément à un ensemble;
- Ajout d'un élément à un ensemble;
- Cardinalité d'un ensemble.

Les définitions de ces fonctions sont données dans le fichier Python joint.

Ce travail devra être réalisé en équipe de deux (2) mais sous une forme particulière. Il ne sera en effet pas réellement nécessaire de se coordonner lors de ce TP même si cela pourrait avoir quand même de certains avantages notamment sur l'aide mutuelle.

En effet, chacun des membres de l'équipe devra seulement développer la moitié des fonctions demandées mai devra également faire une **fonction de test pour chacun des fonctions faite par son coéquipier.** Chacun aura ainsi à cœur la réalisation et la validation du module.

Voici le découpage des fonctions à développer pour chacun des étudiants de l'équipe :

Étudiant 1	**Étudiant 2**
Appartenance	Cardinalité
Intersection	Union
Différence	Différence symétrique
Ajout	Retrait
Test de la cardinalité	Test de l'appartenance
Test de l'union	Test de l'intersection
Test de la différence symétrique	Test de la différence
Test du retrait	Test de l'ajout

Contrainte supplémentaire

Les ensembles seront représentés par des listes Python. Les paramètre de ces fonctions seront donc majoritairement des listes.

Tests unitaires des fonctions

La validité des fonctions et leur efficacité ne sera pas la seule chose évaluée durant ce TP. L'exhaustivité des tests effectués sera également prise en compte.

Un bon test unitaire est un test qui vérifie si non seulement la fonction effectue le travail demandé dans le cas général avec des entrées quelconques, mais vérifie aussi sa validité lors de l'utilisation avec des cas particuliers.

Barème utilisé

Voici le barème (non-détaillé) qui sera utilisé lors de la correction de ce TP.

Fonction		Valeur
Appartenance		5
Cardinalité		5
Intersection		10
Union		10
Différence		10
Différence symétrique		10
Ajout		5
Retrait		5
Test Appartenance		4
Test Cardinalité		4
Test Intersection		4
Test Union		4
Test Différence		4
Test Différence symétrique		4
Test Ajout		4
Test Retrait		4
Remise en avance	**1% par jour d'avance**	**4**
Tests Automatiques	**1% par test automatique passé**	**8**
Total		**104**

Remarquez que, d'une part, 46% de la note est attribuée à chacun des étudiants et que d'autre part, 8% de la note finale sera attribuée automatiquement par un test global scripté fait par nos soins. Il consistera à un test unitaire de chacune des fonctions à développer.
Pour vous assurer que les tests automatiques fonctionneront, vous devez utiliser et compléter les fichiers fournis avec cet énoncé en veillant à ne pas modifier les noms de fonctions et les paramètres de ces fonctions. Le non-respect de cette consigne pourrait vous coûter jusqu'à 8% de votre note.

Remarquez enfin que 4% de bonus sont attribués si vous remettez le travail en avance. **La note finale ne pourra cependant pas excéder 100%.**