Projet Python
Système de recommandation de livres

Consignes & Informations générales :

⇒ Ce projet est à réaliser exclusivement en langage Python

⇒ Pièces jointes au sujet du projet :
- Liste initiale des livres dans le dépôt **books.txt**
- Pour la partie III : les fichiers **readers.txt** et **booksread.txt**
- Pour la partie III : La matrice de notation correspondante aux fichiers fournis

⇒ Organisation des équipes :
- Ce projet est à réaliser en binôme (**un seul trinôme est autorisé** si un **nombre impair** d'élèves)
- La liste des équipes est à remettre aux enseignants **au plus tard** à la fin de la première séance de suivi de projet

⇒ Dates clé :
- Date de publication : **08/11/2021**
- Date de présentation du projet : **10/11/2021**
- Date de suivi 1 : Semaine du **15/11/2021**
- Date de suivi 2 : Semaine du **06/12/2021**
- Date de soumission : **19/12/2021** à **23h59**
- Date de soutenance : Semaine du **03/01/2022**

⇒ Rendu final : Une archive **.zip** contenant
- Le code du projet contenant les fichiers **.py** et **.txt**
- Le rapport en **.pdf**
- Un fichier **README.txt** donnant la liste des programmes et comment les utiliser en pratique. Ce fichier doit contenir toutes les instructions nécessaires à l'exécution ; il est très important pour expliquer à l'utilisateur comment se servir de l'outil.

⇒ Dépôt du projet :
- Sera communiqué plus tard …

⇒ Évaluation
- Barème indicatif
- Barème détaillé : sera fourni plus tard
- Note finale du projet = Note code + Note rapport + Note soutenance
- Rappel : note projet = 20% de la note du cours « Programmation Python »
- Les membres d'une même équipe peuvent avoir des notes différentes en fonction des efforts fournis dans la réalisation de ce projet.

⇒ Plagiat
- Tout travail présentant du plagiat sera sévèrement sanctionné

Organisation du code :

⇒ La notation du code tiendra compte principalement de :
- L'implémentation des fonctionnalités demandées : avancer au mieux mais PAS en hors sujet
- La qualité du code fourni : organisation en fonctions, **commentaires**, noms de variables significatifs, respect des noms de fichiers.
- Facilité d'utilisation de l'interface utilisateur

Notions pédagogiques traitées :

⇒ La réalisation de ce projet vous aidera à appliquer les notions pédagogiques suivantes :
- Notions des bases, listes 1D, listes 2D, les fonctions et les fichiers

⇒ Pour vous aider à la réalisation de ce projet, vous pouvez vous appuyer sur :
- Les supports du cours TI101 (I, B, R)
- Livres, cours divers sur le web. **ALERTE PLAGIAT !!** il ne s'agit pas de copier des programmes entiers.
- Enseignants **Efrei** lors des séances de suivi du projet

Préambule

Le but de ce projet est de réaliser un outil informatique permettant de suggérer (recommander) aux lecteurs des livres en fonction de leurs profils et de leurs anciennes lectures.

Pour atteindre cet objectif, ce projet est divisé en 3 parties :

- Partie 1 : Gestion des profils des lecteurs
    - Ajouter un lecteur
    - Afficher un lecteur
    - Modifier un lecteur
    - Supprimer un lecteur

- Partie 2 : Gestion du dépôt de livres
    - Afficher la liste des livres dans le dépôt (liste initiale fournie dans le fichier **books.txt**)
    - Ajouter un livre au dépôt
    - Modifier le titre d'un livre dans le dépôt
    - Supprimer un livre du dépôt

- Partie 3 : Recommandation d'un livre
    - Noter un livre
    - Suggérer des livres

Le lancement du programme doit donc afficher à l'utilisateur un menu pour choisir entre 3 fonctionnalités qui représentent chacune des parties décrites précédemment :

1. Profils des lecteurs
2. Visiter le dépôt des livres
3. Recommandation

Partie I : Profils des lecteurs (8 pts)

Les paramètres liés au profil d'un lecteur seront au minimum (vous pouvez en ajouter d'autres en fonction de l'utilité et de votre imagination) :

- Le pseudonyme
- Le genre
    1. HOMME.
    2. FEMME
    3. PEU IMPORTE

- L'âge
    1. <= 18 ans
    2. Entre 18 ans et 25 ans
    3. > 25 ans
- Le style de lecture
    1. Science-fiction
    2. Biographie
    3. Horreur
    4. Romance
    5. Fable
    6. Histoire
    7. Comédie
- La liste des livres lus parmi ceux présents dans le dépôt

Toutes ces informations seront stockées dans des listes et fichiers pour pouvoir les attribuer aux profils et/ou les afficher aux utilisateurs.

Représentation et stockage des profils des lecteurs

Tous les profils des lecteurs seront stockés dans deux fichiers texte contenant une ligne par lecteur inscrit :

- Dans le fichier **readers.txt**, le lecteur sera stocké en une seule ligne sous le format suivant :

  Pseudo, **numéro genre**, **numéro age**, **numéro style lecture**

  Exemple :

  La ligne suivante du fichier **readers.txt** :

  Casanova, **1**, **2**, **6**

  Signifie que le lecteur s'est inscrit sous le pseudonyme **Casanova**, c'est un **HOMME**, âgé **entre 18 et 25 ans** et qui aime lire des livres d'**Histoire**.

- Dans le fichier **booksread.txt**, une ligne sera consacrée à chaque lecteur inscrit qui sera sous la forme :

  Pseudo, **numéro livre lu 1, numéro livre lu 2, …**

  En effet, lors de la phase de saisie du profil du lecteur, le programme doit lui afficher la liste des livres se trouvant dans le dépôt (fichier **books.txt**) et l'utilisateur saisit les numéros des livres qu'il a déjà lus et qui seront stockés sur la même ligne que son pseudo dans le fichier **booksread.txt.**

  Exemple :

  Pour cette liste des livres :

  ```
  1 - Débuter la programmation en langage Java
  2 - Apprendre Python
  ```

```
3 - Les Citations du Président Mao Tse-Toung
4 - Don Quichotte de la Manche
5 - Un conte de deux villes
6 - Le Seigneur des Anneaux
7 - Le Petit Prince
```

La ligne suivante du fichier **booksread.txt** :

Casanova, 1, 4, 6

Signifie que le lecteur **Casanova** a déjà lu les livres :

- **Débuter la programmation en langage Java**
- **Don Quichotte de la Manche**
- **Le Seigneur des Anneaux**

Fonctionnalités à réaliser :

1. **Afficher les livres** : Chaque nouveau lecteur doit pouvoir consulter la liste des livres se trouvant dans le dépôt pour sélectionner ceux qu'il aurait déjà lus.

   Entrée : le fichier **books.txt**

   Sortie : affichage de la liste des livres du fichier **books.txt** précédé chacun par un numéro séquentiel.

2. **Ajout d'un lecteur** : un nouveau lecteur doit pouvoir saisir son profil en répondant successivement aux questions permettant de renseigner les informations décrites ci-dessus.

   Entrée : les fichiers **readers.txt** et **booksread.txt**

   Sortie : Les mêmes fichiers enrichis chacun d'une nouvelle ligne.

   Démarche : Pour les champs ayant des valeurs précises, effectuer la saisie sécurisée pour que seules ces valeurs soient entrées par l'utilisateur. L'utilisateur doit pouvoir saisir plusieurs livres qu'ils auraient déjà lus ➔ Lui afficher la liste de tous les livres pour chaque saisie.

3. **Afficher un lecteur** : l'utilisateur doit pouvoir afficher le profil d'un lecteur donné.

   Entrée : les fichiers **readers.txt** et **booksread.txt**

   Sortie : Affichage des informations (compréhensibles par un humain) de ce lecteur.

4. **Modifier un lecteur** : l'utilisateur doit pouvoir modifier les informations du profil d'un lecteur donné.

   Entrée : les fichiers **readers.txt** et **booksread.txt**

   Sortie : les fichiers **readers.txt** et **booksread.txt** mis à jour.

5. **Supprimer un lecteur** : l'utilisateur doit pouvoir supprimer le profil d'un lecteur donné.

   Entrée : les fichiers **readers.txt** et **booksread.txt**

   Sortie : Les mêmes fichiers, amputés de la ligne concernant le lecteur spécifié.

Partie II : Visiter le dépôt des livres (6 pts)

Le fichier **books.txt** contient déjà une liste initiale de livres à lire. Pour des raisons de simplicité, nous nous contentons de définir chaque livre par son titre. Chaque titre est stocké sur une ligne du fichier. Nous pouvons déjà afficher cette liste grâce à la fonctionnalité « **afficher les livres** » présente dans la partie I.

Autres fonctionnalités à réaliser :

1. **Ajouter un livre** : Chaque utilisateur doit pouvoir ajouter un titre de livre au dépôt mais seulement si celui-ci n'existait pas avant. Si l'ajout s'effectue, le titre sera ajouté automatiquement à la fin du fichier.

   Entrée : le fichier **books.txt**

   Sortie : Le fichier **books.txt** enrichi d'un nouveau titre de livre.

2. **Modifier un livre** : Chaque utilisateur doit pouvoir modifier le titre d'un livre du dépôt seulement si celui-ci existe déjà.

   Entrée : le fichier **books.txt**

   Sortie : Le fichier **books.txt** mis à jour.

3. **Supprimer un livre** : Chaque utilisateur doit pouvoir supprimer un livre du dépôt seulement si celui-ci existe déjà. Avant la suppression, il est nécessaire de repérer son rang dans la liste des livres (**books.txt**) et de supprimer ce numéro sur toutes les lignes des lecteurs dans le fichier **booksread.txt** .

   Entrée : les fichier **books.txt** et **booksread.txt**

   Sortie : Le fichier **books.txt** amputé d'une ligne et le fichier **booksread.txt** mis à jour.

Partie III : Recommandation (6 pts)

Le système de suggestion à développer ici se base sur les avis des lecteurs. En effet, pour pouvoir suggérer un livre à un lecteur, il est essentiel que ces derniers déposent des notes pour les livres lus. Les valeurs des notes vont de 1 (je n'ai pas aimé) à 5 (excellent).

Cette opération donne naissance à une matrice de notation dont les lignes représentent les lecteurs et les colonnes représentent les livres présents dans le dépôt.

Matrice de notation

L'obtention de cette matrice et la garantie de sa cohérence nécessitent l'amélioration de quelques-unes des fonctionnalités précédentes.

Fonctionnalités à réaliser :

1. **Créer la matrice de notation** : Dès le lancement du programme la matrice de notation doit être créée et initialisée à 0.

   Entrée : Les fichiers **books.txt** (nombre de livres) et **readers.txt** (nombre de lecteurs)

   Sortie : La matrice de notation créée et initialisée à 0.

2. **Noter un livre** : Chaque utilisateur doit pouvoir noter un livre à condition qu'il l'ait déjà lu. Cette opération peut se faire soit à l'ajout d'un nouveau lecteur ou lorsque l'utilisateur choisit lui-même de noter un livre.

   Entrée : La matrice de notation, le pseudonyme du lecteur (optionnel), les fichiers **readers.txt**, **books.txt** et **booksread.txt.**

   Sortie : La matrice de notation mise à jour.

   Démarche :

   - Cette fonction va demander à l'utilisateur de saisir son pseudo si elle n'est pas appelée depuis la fonction d'ajout d'un lecteur.
   - Vérifie si le pseudonyme existe bien dans le fichier « **readers.txt** »
   - Demande à l'utilisateur de saisir le titre du livre qu'il souhaite noter
   - Vérifie si le titre existe bien dans le fichier « **books.txt** ». Si oui, elle récupère sans rang.
   - Vérifier dans le fichier « **booksread.txt** » si le lecteur en question a bien lu le livre.
   - Si c'est le cas, elle lui permet de noter le livre en s'assurant que la note soit bien un entier entre 1 et 5.

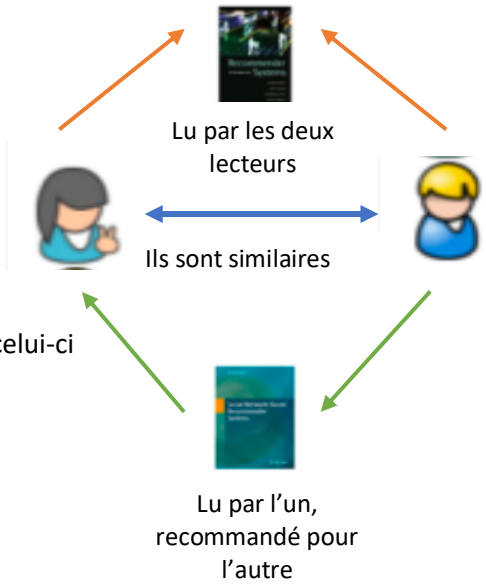
Fonctionnalités à mettre à jour :

1. **Ajouter un lecteur** : A chaque ajout d'un lecteur dans le fichier « **readers.txt** », une ligne est ajoutée à la matrice de notation où toutes les valeurs sont à 0.
2. **Supprimer un lecteur** : A chaque suppression d'un lecteur du fichier « **readers.txt** », la ligne correspondant à son rang est supprimée de la matrice de notation.
3. **Ajouter un livre** : A chaque ajout d'un livre dans le fichier « **books.txt** », une colonne doit être ajouté à la matrice de notation où toutes ses valeurs sont à 0.

4. **Supprimer un livre** : A chaque suppression d'un livre du fichier « **books.txt** », la colonne correspondant à son rang est supprimée de la matrice de notation.

Suggestion de livres

L'algorithme de recommandation à implémenter ici se base sur la similarité des profils des lecteurs. Autrement dit, si deux lecteurs l1 et l2 ont lu des livres en commun, le système les verra comme ayant des profils similaires. Ainsi, si l'un d'entre eux (soit l1) a lu un livre supplémentaire, celui-ci sera recommandé au lecteur l2.

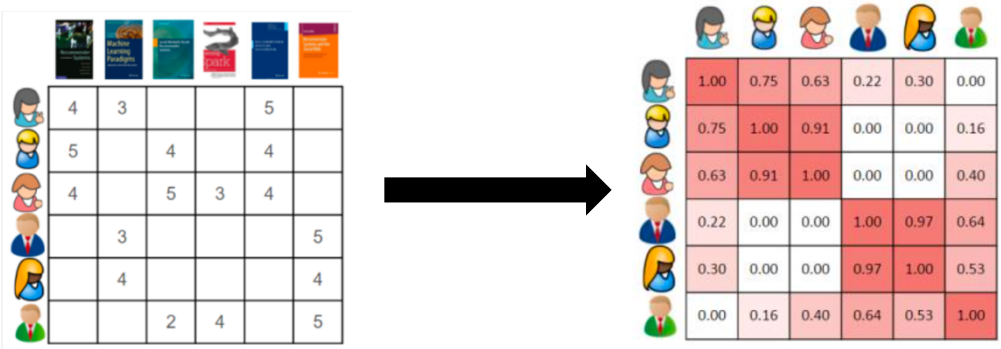


La méthode s'applique comme suit :

- À partir de la matrice de notation, calculer une matrice de similarité
- La matrice de similarité est une matrice carrée dont le nombre de lignes = nombre de colonnes = nombre de lecteurs dans le fichier « **books.txt** ».
- Chaque case de la matrice de similarités représente le taux de similarité d'un lecteur avec un autre. Ce taux appartient à l'intervalle [0, 1].
- La diagonale de la matrice de similarité est toujours à 1 car tout lecteur est exactement similaire à lui-même.
- La fonction de similarité à utiliser ici est appelée « La similarité Cosinus » et qui se calcule comme suit :
  En considérant 2 vecteurs $A = [a_1, a_2, \ldots, a_n]$ et $B = [b_1, b_2, \ldots, b_n]$ à **n** attributs chacun, la similarité de cosinus de ces deux vecteurs se calcule comme suit :

$$\frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \; x \; \sqrt{\sum_{i=1}^{n} b_i^2}}$$

- Dans notre cas, chaque ligne de la matrice de notation représentera un vecteur. Ainsi, en considérant une ligne i_1 comme le premier vecteur, et une ligne i_2 comme le deuxième vecteur, cela revient à calculer la similarité entre le lecteur du rang i_1 et le lecteur de rang i_2.



- Une fois la matrice de similarité calculée, il suffira de :
    1. Considérer un lecteur (une ligne de la matrice de similarités) : soit **i**
    2. Voir avec qui il est le plus similaire (la valeur max sur la ligne sélectionnée) et identifier sa colonne : soit **j**
    3. Consulter dans le fichier « **booksread.txt** » les livres lus par le lecteur **j** et non lus par le lecteur **i** et les suggérer au lecteur **i**.

Fonctionnalités à réaliser :

1. **Suggérer un livre** : Implémenter la méthode de recommandation décrite ci-dessus pour un utilisateur qui demande à lui suggérer des livres.

   Entrée : Les fichiers **readers.txt** et **booksread.txt**, la matrice de notation

   Sortie : Le fichier **booksread.txt** mis à jour

   Démarche :
   - Demander à l'utilisateur de saisir son pseudonyme.
   - Vérifier qu'il existe bien dans le fichier « **booksread.txt** »
   - Calculer la matrice de similarité.
   - Lui suggérer tous les livres lus par le lecteur qui lui ressemble le plus et qu'il n'a pas lu avant.
   - Lui proposer de sélectionner un titre
   - Lorsque le titre est sélectionné, son rang est ajouté dans la liste des livres lus pour l'utilisateur en question dans le fichier « **booksread.txt** »
   - Proposer au lecteur s'il souhaite noter le livre sélectionné (en supposant qu'il l'a déjà lu).
   - En cas de refus, lui afficher un message lui rappelant de penser à noter son livre après lecture.
   - Pour aller plus loin : calculer le temps de calcul de la matrice de similarité.

NB : Pour obtenir des résultats intéressants lors de vos tests, n'hésitez pas à utiliser les fichiers « readers.txt » et « booksread.txt » ainsi que la matrice de notation fournis avec le sujet de ce projet.

Remarques générales :

⇒ Les saisies sécurisées sont à effectuer systématiquement même si elles ne sont pas toujours demandées explicitement
⇒ Ne pas hésiter à afficher des messages aux utilisateurs lorsqu'une action n'est pas possible.
⇒ Afficher le menu à la fin de chaque action pour pouvoir basculer vers une autre action.