



## Les fichiers d'entités

### TD5

#### Exercice 1

Une société de distribution d'eau dispose d'un fichier contenant toutes les factures de ses clients. Soit l'entité 'Facture' contenant l'identifiant et le nom du client ainsi que le montant à payer et l'état du paiement (=1 si la facture a été payée et =0 si la facture n'a pas été payée).

**entité** Facture  
**composantes:**

IdC: **entier**  
Nom: **chaîne**  
Montant : **réel**  
Paiement : **entier**

**fent**

Soit un fichier d'entités du type 'Facture'. Le fichier est triée **en ordre croissant** sur le champ 'IdC'.

Rédigez en Python un programme

- ✓ qui affiche, pour chaque client, son nom, le montant total à payer
- ✓ qui affiche le nom du client devant payer la plus petite somme totale (qu'elle soit déjà payée ou pas)
- ✓ qui affiche le nombre de factures impayées à ce jour
- ✓ qui affiche le nombre total de clients différents présents dans la liste de clients
- ✓ qui affiche la somme totale de toutes les factures, payées ou pas.

#### EXEMPLE (PEU RÉALISTE)

Fichier de clients : 1/A/100/1; 1/A/300/1; 1/A/40/1; 3/D/100/0;  
3/D/120/1; 3/D/80/1; 3/D/40/1; 6/M/100/0; 6/M/100/0; 9/T/120/1

Le client A doit payer 440 €

Le client D doit payer 340 €  
Le client M doit payer 200 €  
Le client T doit payer 120 €

Il y a 4 clients différents  
Le client T doit payer le plus petit montant total (120 €)  
Le nombre de factures pour lesquels les clients n'ont pas payé dans les délais = 3  
La somme totale des factures est de 1100 €

## Notes

1. Le fichier ne peut être lue qu'une seule fois et seulement d'une manière séquentielle à partir du début du fichier.
2. Une fois que le fichier est constitué, le fichier ne peut pas être stocké dans un (des) tableau(x) ou structure(s) similaire(s).
3. Le fichier peut être vide.
4. L'affichage peut être différent de celui de l'exemple.
5. Vous ne pouvez pas utiliser les structures de contrôles et autres fonctionnalités du langage Python qui n'ont pas été vues en classe.

## Exercice 2

Soit l'entité Eleve

**entité** Eleve

composantes : Id : **entier**

Nom : **chaîne**

Note : **entier**

**fent**

Rédigez un programme qui pour tous les élèves apparaissant dans un fichier, affiche son nom et sa moyenne (somme des notes divisée par le nombre de notes). Le programme doit également afficher le nombre total d'élèves et le nombre d'élèves ayant obtenu plus de 10 de moyenne (il est donc supposé que chaque note est sur 20 et que les notes ont le même poids).

### EXEMPLE (PEU RÉALISTE)

Fichier d'élèves : 1/A/10; 1/A/15; 1/A/6; 2/B/16; 2/B/11; 3/C/3;  
5/E/11; 8/H/14; 8/H/19; 9/I/20

Elève A moyenne de : 10.33  
Elève B moyenne de : 13.5  
Elève C moyenne de : 3  
Elève E moyenne de : 11  
Elève H moyenne de : 16.5  
Elève I moyenne de : 20

Il y a 6 élèves différents

5 élèves ont obtenus une moyenne de plus de 10

### Notes

1. Le fichier ne peut être lue qu'une seule fois et seulement d'une manière séquentielle à partir du début du fichier.
2. Une fois que le fichier est constitué, le fichier ne peut pas être stocké dans un (des) tableau(x) ou structure(s) similaire(s).
3. Le fichier peut être vide.
4. L'affichage peut être différent de celui de l'exemple.
5. Vous ne pouvez pas utiliser les structures de contrôles et autres fonctionnalités du langage Python qui n'ont pas été vues en classe.

### Exercice 3

Une compagnie aérienne propose à ses clients des voyages gratuits en fonction des kilomètres parcourus à bord de ses avions.

Soit l'entité '**Voyageur**' contenant le numéro du client, son nom ainsi que le nombre de kilomètres parcourus lors du voyage correspondant à l'enregistrement.

#### entité **Voyageur**

composantes : No : entier

Nom : chaîne

Km : réel

#### fent

Le fichier du type '**Voyageur**' contient tous les enregistrements générés à chaque nouveau voyage des clients. La liste est triée en ordre croissant sur le champ 'No' et respecte l'ordre chronologique des voyages.

Rédigez un programme qui, pour chaque Voyageur, affiche son nom et le nombre total de kilomètres parcourus. Le programme affiche aussi le nombre total de voyageurs ayant parcouru 5000 kms ou plus.

### EXEMPLE (PEU RÉALISTE)

Fichiers de voyageurs : 1/A/1000; 2/B/100; 2/B/5000; 4/D/1600;  
6/F/3500; 7/G/3000; 7/G/1100; 8/H/4000; 8/H/1900; 9/I/2000

Voyageur A total de : 1000.0

Voyageur B total de : 5100.0

Voyageur D total de : 1600.0

Voyageur F total de : 3500.0

Voyageur G total de : 4100.0

Voyageur H total de : 5900.0

Voyageur I total de : 2000.0

2 voyageurs ont fait plus de 5000Km

## Notes

1. Le fichier ne peut être lue qu'une seule fois et seulement d'une manière séquentielle à partir du début du fichier.
2. Une fois que le fichier est constitué, le fichier ne peut pas être stocké dans un (des) tableau(x) ou structure(s) similaire(s).
3. Le fichier peut être vide.
4. L'affichage peut être différent de celui de l'exemple.
5. Vous ne pouvez pas utiliser les structures de contrôles et autres fonctionnalités du langage Python qui n'ont pas été vues en classe.

## Exercices d'auto-évaluation :

### Exercice 1

Une association à but non lucratif a récolté des dons en argent afin d'offrir des aides ponctuelles aux enfants et aux jeunes dans le besoin, et ce tout au long de l'année 2020. Pour faire le bilan annuel des dons par donateur, vous disposez de 2 fichiers : un fichier contenant les renseignements concernant les différents donateurs et un autre avec les dons donnés.

Soient l'entité 'Don' contenant l'identifiant d'un donateur son nom et la somme donnée.

**entité** Don

**composantes** : IdDonateur : **entier**

Nom : **chaîne**

Somme: **entier**

**fent**

Rédigez en Python un programme :

- ✓ qui affiche le nom de chaque donateur suivi de la somme totale qu'il a donnée au cours de l'année 2020 et du nombre de dons ;
- ✓ qui affiche le nombre de donateurs différents ;
- ✓ qui affiche le nom du donateur ayant versé la plus grande somme ;
- ✓ qui affiche la somme moyenne donnée (=somme totale donnée par tous les donateurs / le nombre de donateurs différents) ;

### EXEMPLE (PEU RÉALISTE)

Fichier de Dons : **1/N/100; 1/N/150; 1/N/200; 3/B/250; 5/A/300; 7/Y/120; 9/C/100; 9/C/120; 9/C/80**

Le donateur N a donné 450 € (3 dons)

Le donateur B a donné 250 € (1 don)

Le donateur A a donné 300 € (1 don)

Le donateur Y a donné 120 € (1 don)

Le donateur C a donné 300 € (3 dons)  
5 donateurs différents pour l'année 2020  
N est le donateur ayant donné la plus grande somme  
La somme totale donnée = 1420.0 €  
La somme moyenne donnée = 284 €

## Notes

1. Le fichier ne peut être lue qu'une seule fois et seulement d'une manière séquentielle à partir du début du fichier.
2. Une fois que le fichier est constitué, le fichier ne peut pas être stocké dans un (des) tableau(x) ou structure(s) similaire(s).
3. Le fichier peut être vide.
4. L'affichage peut être différent de celui de l'exemple.

## Exercice 2

Un club sportif de football fait appel à vos talents pour faire le bilan annuel des buts marqués par joueur durant l'année 2023. Pour faire cela, vous disposez d'un fichier contenant les renseignements concernant les différents joueurs ainsi que les buts marqués.

Soient l'entité 'Joueur' contenant l'identifiant d'un joueur, son nom ainsi que le nombre de buts marqués par match.

**entité** Joueur

**composantes** : IdJ: **entier**

Nom : **chaîne**

NbreBut: **entier**

**fent**

Soient "FJoueurs" un fichier d'entités du type 'Joueur' contenant les renseignements concernant tous les joueurs et buts marqués. Le fichier est trié en ordre croissant sur le champ 'IdJ'.

Rédigez en Python un programme :

- ✓ qui affiche le nom de chaque joueur suivi du nombre total de buts marqués ainsi que le nombre de matchs dans lesquels il a joué  
Attention :  
➔ Si le nombre total de buts marqués est nul, vous devez afficher le nom du joueur et le message "aucun but"
- ✓ qui affiche le nombre de joueurs différents qui affiche le nombre de joueur n'ayant pas marqué de buts
- ✓ qui affiche le nom du premier joueur ayant marqué le plus de buts
- ✓ qui affiche le nombre total de buts marqués

#### EXEMPLE (PEU RÉALISTE)

Fichier de Joueurs : **1/N/2;1/N/2; 1/N/3;2/F/0; 3/B/1; 3/B/2; 4/M/0; 5/A/1; 7/Y/2; 9/C/0; 9/C/2; 9/C/2, 10/X/0;**

Le joueur N a marqué 7 buts (Le nombre de matchs = 3)  
Le joueur F : aucun but (Le nombre de matchs = 1)  
Le joueur B a marqué 3 buts (Le nombre de matchs = 2)  
Le joueur M : aucun but (Le nombre de matchs = 1)  
Le joueur A a marqué 1 but (Le nombre de matchs = 1)  
Le joueur Y a marqué 2 buts (Le nombre de matchs = 1)  
Le joueur C a marqué 4 buts (Le nombre de matchs = 3)  
Le joueur X : aucun but (Le nombre de matchs = 1)

8 joueurs différents  
3 joueurs n'ont pas marqué de buts durant l'année 2023  
N est le joueur ayant marqué le plus de buts (7 buts)  
Le nombre total de buts = 17

#### Notes

1. Le fichier ne peut être lue qu'une seule fois et seulement d'une manière séquentielle à partir du début du fichier.
2. Une fois que le fichier est constitué, le fichier ne peut pas être stocké dans un (des) tableau(x) ou structure(s) similaire(s).
3. Le fichier peut être vide.
4. L'affichage peut être différent de celui de l'exemple.