

R1.04 - Systèmes - TP 5

Filtres (suite)

Mise en pratique

Les exercices suivants vont vous permettre de mettre à l'épreuve votre réflexion dans la mise en œuvre des différents filtres que vous venez de voir dans la 1^{ère} partie du TP 5 (Filtres), afin de produire le résultat attendu.

N'oubliez pas la finalité des filtres : créer des chaînes de transformation des données.

Sauf indication contraire, les commandes ou combinaisons de commandes doivent afficher leur résultat à l'écran (**STDOUT**).

En aucun cas il ne faut modifier les fichiers de travail fournis avec ce TP. Si vous les modifiez par erreur (un > au lieu d'un < par exemple), repartez des originaux de Moodle.

Quand on vous écrit :

uniquement ou ***unique***,

il ne faut pas afficher les lignes entières, mais uniquement les informations indiquées. Dans ce cas, vous pouvez aussi conserver le caractère séparateur à l'affichage.

Quand on vous écrit :

le fichier ou ***les lignes*** ou rien qui suggère uniquement certains champs,

il faut afficher les lignes entières.

Les solutions ne sont pas forcément uniques. Vous pouvez trouver des combinaisons de commandes différentes qui aboutissent au même résultat. Par exemple, choisir d'abord toutes les pommes parmi des fruits et n'en conserver que les rouges est équivalent à choisir d'abord toutes les choses rouges et ensuite n'en garder que les pommes.

Sauf indication contraire, les questions sont indépendantes.

La complexité des questions est variable. Laissez de côté celles sur lesquelles vous butez pour y revenir dans un second temps.

Conseil-rappel : n'hésitez pas à y aller par étape si vous avez plusieurs enchaînements de commandes (et il en aura beaucoup !), pour voir à l'écran, à chaque étape, ce que ça produit avant de vous engager dans l'écriture de la suite de la commande.

Fichier prod

Affichez :

- Q.1 Chaque ligne convertie en majuscules.
- Q.2 Les produits et les quantités de chaque ligne (uniquement ces 2 champs), et triez par quantité.
- Q.3 Les couleurs uniques, et pour chacune des couleurs, le nombre de fois qu'elles sont présentes. Seules ces 2 informations seront affichées, et sans formatage particulier.

Fichier lang

Affichez :

- Q.4 Le fichier en supprimant les espaces redondants (on n'en garde qu'un seul à chaque occurrence multiple). NB : supprimer ne signifie pas remplacer par autre chose.
- Q.5 Le fichier trié sur l'année, puis sur le nom (appuyez-vous sur la question précédente).
- Q.6 Les années uniques, sans le siècle (juste les 2 derniers chiffres).
- Q.7 Le nombre de langages dont le nom de ne dépasse pas 3 lettres.

Fichier depts

Affichez :

- Q.8 Chaque ligne en remplaçant les apostrophes par des espaces (piège).
- Q.9 La liste des initiales uniques des départements. Ça pourrait servir pour identifier les lettres qui n'ont aucun département commençant par cette lettre. Quelles lettres n'ont pas de département correspondant ? (Ce n'est pas la commande qui vous donne la réponse mais le complément de ce qui s'affiche).

- Q.10 En vous inspirant de la question précédente, affichez pour chaque lettre le nombre de départements commençant par cette lettre, et triez par ordre décroissant de nombre. Quelle lettre possède le plus de départements ayant cette initiale ?

Fichier murphy

Affichez :

- Q.11 Le fichier en remplaçant chaque espace par un `_` (underscore).
- Q.12 Le fichier en supprimant les espaces.
- Q.13 Le fichier en supprimant les `;` (points-virgules) (piège).

Fichier lorem

Affichez :

- Q.14 Le fichier en supprimant les voyelles.
- Q.15 Le fichier en convertissant les minuscules en MAJUSCULES et inversement.

Les questions suivantes sont un exercice de synthèse qui va vous guider pas à pas vers le résultat final, à savoir : quels sont les 3 mots les plus fréquents du fichier **lorem** ?

Pour info, ce sont les mots **dolore**, **fugiat** et **irure**

Pour y parvenir, suivez ces étapes :

- Q.16 Affichez le fichier en remplaçant les retours à la ligne par des espaces.
- Q.17 Améliorez la commande précédente en ne gardant qu'un seul espace partout où il y en a plusieurs.
- Q.18 A partir de la commande précédente supprimez les `.` (points).

- Q.19 A partir de la commande précédente supprimez les . (points), affichez tous les mots, un par ligne.
- Q.20 A partir de la commande précédente, listez chaque mot et le nombre de fois où il apparaît.
- Q.21 A partir de la commande précédente, affichez les 3 mots les plus fréquents (sans supprimer l'information de comptage).
- Q.22 A partir de la commande précédente, nettoyez l'affichage pour ne garder que les 3 mots les plus fréquents. Obtenez-vous le résultat attendu (voir la liste en introduction de cet exercice) ?
- Q.23 En observant le contenu du fichier **lorem**, on observe que certains mots (ex : **fugiat**) sont présents sous deux écritures différentes. Si vous n'en avez pas déjà tenu compte, corrigez votre commande pour ne pas faire de différence entre ces écritures. En principe il n'y aura plus qu'un seul mot à apparaître le plus souvent.

Code César

Le célèbre Jules César, empereur romain du 1^{er} siècle avant JC, a imaginé un code de brouillage des ordres qu'il envoyait à ses troupes par messagers. Le principe est simple, mais était suffisamment efficace pour l'époque. Il consiste en un décalage de toutes les lettres du message, d'un certain nombre de lettres vers la gauche ou vers la droite. Exemple avec un décalage de 1 caractère à droite :

A devient **B**, **B** devient **C**, **C** devient **D**, etc. en finissant par **Z** devient **A**

Ainsi, avec le mot **OISEAU**, le **O** devient **P**, le **I** devient **J**, et ainsi de suite pour produire **PJTFBV**.

Pour décoder, il suffit de faire l'inverse (décalage de 1 caractère à gauche dans notre exemple).

- Q.24 Proposez une commande permettant de passer en majuscules le contenu du fichier **clair** pour produire un fichier nommé **cesar**.

Pour les questions suivantes, vous partirez de ce fichier **cesar**.

- Q.25 Proposez une commande permettant de coder le contenu du fichier **cesar** par un décalage de 1 caractère vers la droite. Vous stockerez le résultat du codage dans un fichier nommé **crypto**.
- Q.26 Proposez une autre commande permettant de décoder le contenu du fichier **crypto** par un décalage de 1 caractère vers la gauche. Vérifiez que le résultat obtenu est de nouveau lisible et identique au contenu du fichier **clair**.
- Q.27 Proposez une commande permettant de coder le contenu du fichier **cesar** par un décalage de **13** caractères vers la droite. Vous stockerez le résultat du codage dans un fichier nommé **crypto13**.
- Q.28 Essayez de coder le fichier **crypto13** (qui est déjà codé) par la même technique de codage qu'à la question précédente. Qu'observez-vous ? Comprenez-vous pourquoi ?
- Q.29 Avec ce décalage de **13**, a-t-on toujours besoin d'une commande de décodage ? Est-ce que ça fonctionnerait avec l'alphabet russe¹ ?
- Q.30 Testez votre commande précédente sur le fichier **proverbes** pour découvrir le message caché.

¹ https://fr.wikipedia.org/wiki/Alphabet_russe
