
Scala -- Programmation fonctionnelle –**TP 5 : Mapping /Filtering/Reducing**

Exercice 01 :

Soit la liste suivante : List(« Bonjour», « Aujourd’hui», « C’est», « La », « Fete », « Des », « Papas»)

En utilisant les fonctions vues en cours, écrire un code scala qui permet de :

- retourner une nouvelle liste, ou chaque élément représente la longueur du mot à la même position.
- Retourner une nouvelle liste, ou chaque élément est un couple de valeurs (mot, longueur_mot).

Exercice 02 :

Soit la Map suivante : Map(1 -> "un", 2 -> "Deux", 3 -> "Trois", "Quatre", 5 -> "Cinq", 6-> "Six");

- Ecrire une programme scala qui permet d’écrire les valeurs de la Map en majuscule.

Exercice 03 :

Soit l'instruction suivante : **val** seq = 0 until 7 ;. Elle est l'équivalent de 0, 1, 2, 3, 4, 5, 6 (sans le 7)

- Ecrire un programme scala qui retourne une collection (vecteur) seq1, tel que chaque élément est 0 until n, avec n une valeur de seq

i.e : (0 until 7, 1 until 7, 2 until 8, ...)

- Puis transformer ce vecteur en une liste de valeur composée de toutes les valeurs seq1.

i.e : (0, 1, 2, ...7, 1, 2, ...7, 2, 3, ..., 7, 3,..., 7, ...).

Exercice 04 :

Soit la liste suivante : **val** liste = List("10", "52", "65", "85", "96", "42", "88", "32", "83", "122", "823", "786", "95", "81", "87", "13", "19", "78", "65", "485", "62", "55", "292", "472", "712");

- Avec une seule instruction, calculer la somme des nombres paires de cette liste.

Exercice 05 :

Soit le fichier table.txt qui contient une ligne entête et un ensemble de tuples, ou chaque tuple est une affectation de valeur pour les variables d'entête.

Le but de l'exercice est de garder dans la table uniquement les items les plus fréquents :

- On appelle un item la paire (variable, valeur).
- Un item est fréquent son nombre d'occurrence est supérieur ou égal à un seuil minimal (S_{min}).

Pour se faire :

- Lire le fichier table.txt,
- Ecrire les tuples sous forme d'une séquence d'items. l
 - Exemple : le premier tuple sera écrit comme suit :
`Seq("x1=0", "x2=1", "x3=2", "x4=1", "x5=2", "x6=3", "x7=4", "x8=2", "x9=1")`
- Calculer le nombre d'occurrences de chaque item dans la séquence précédente,
- Dans une Liste, ordonner chaque tuple de table.txt dans l'ordre décroissant de nombres d'occurrence des items.
- Supprimer de chaque tuple les items ayant un nombre d'occurrences inférieur ou égale au S_{min} . (on considère la valeur de $S_{min} = 3$).