

# TD noté, rattrapage 2009

*Le programme construit au fur et à mesure des questions devra être imprimé à la fin du TD et rendu. Toutes les questions valent 3 points sauf la première et la dernière qui valent 1 point. Tout document autorisé.*

Cet exercice propose de fusionner des données provenant de deux sources. La première étape consiste à télécharger deux fichiers disponibles à l'adresse suivante : <http://www.xavierdupre.fr/mywiki/InitiationPython> :

td\_note\_2009\_cluedo\_1.txt et td\_note\_2009\_cluedo\_2.txt.

Il est préférable de placer ces pièces jointes dans le même répertoire que le programme lui-même. Le premier fichier contient une matrice dont chaque ligne contient 4 informations séparées par des tabulations :

un nom   un prénom   une couleur   une arme

Les premières lignes sont les suivantes<sup>1</sup> :

N10	P10	gris	corde
N15	P15	rouge	chandelier
N8	P8	marron	fer à cheval

Le second fichier contient une matrice dont chaque ligne contient 3 informations séparées par des tabulations :

un nom   un prénom   une pièce

Les premières lignes sont les suivantes<sup>2</sup> :

N3	P3	cuisine
N19	P19	bureau
N20	P20	salle à manger

Certaines élèves sont présents dans les deux fichiers, l'objectif de cet exercice est de construire un troisième fichier regroupant les informations concernant les élèves présents dans les deux fichiers. Ce fichier contiendra donc 5 colonnes.

1) Construire une fonction qui prend comme argument un nom de fichier et qui retourne le contenu du fichier dans une matrice (une liste de listes)<sup>3</sup>.

```
def convertit_fichier_en_matrice (fichier) :  
    ...
```

2) Construire une fonction qui prend comme entrée une matrice et qui construit un dictionnaire dont :

1. Ces données sont complètement aléatoires.
2. Ces données sont aussi complètement aléatoires.
3. On pourra s'inspirer du programme présent à l'adresse [http://www.xavierdupre.fr/enseignement/initiation/initiation\\_via\\_python\\_all/chap7\\_fichiers\\_200\\_.html](http://www.xavierdupre.fr/enseignement/initiation/initiation_via_python_all/chap7_fichiers_200_.html).

- la clé est un tuple composé des valeurs des deux premières colonnes (nom, prénom)
- la valeur est la liste des valeurs des autres colonnes

```
def convertit_matrice_en_dictionnaire (matrice) :
...
```

3) Construire une fonction qui effectue la fusion de deux dictionnaires. Le dictionnaire résultant vérifie :

- chaque clé est présente dans les deux dictionnaires à la fois
- la valeur associée à une clé est la somme des deux listes associées à cette même clé dans chacun des deux dictionnaires.

```
def fusion_dictionnaire (dico1, dico2) :
```

4) Construire une fonction qui convertit un dictionnaire en matrice. C'est la fonction réciproque de la fonction de la question 2. Les deux premières colonnes sont celles de la clé, les suivantes sont celles des valeurs.

```
def convertit_dictionnaire_en_matrice (dico) :
```

5) Ecrire une fonction qui écrit une matrice dans un fichier texte.

```
def convertit_matrice_en_fichier (mat, fichier) :
```

6) En utilisant toutes les fonctions précédentes, écrire une fonction qui effectue la fusion de deux fichiers et qui écrit le résultat dans un autre fichier.

```
def fusion_fichier (fichier1, fichier2, fichier_resultat) :
```

7) En vous inspirant du schéma proposé dans les précédentes questions, il s'agit ici de produire un troisième fichier qui contient tous les couples (nom, prénom) qui sont présents dans un fichier mais pas dans les deux à la fois.

```
def union_moins_intersection_fichier (fichier1, fichier2, fichier_resultat) :
```

8) Serait-il compliqué d'écrire une fonction qui permette de fusionner 3 fichiers ?  $n$  fichiers ? Justifier.

**A quoi ça sert :** cette opération de fusion peut être effectuée sous Excel lorsque la taille des fichiers n'excède pas 65000 lignes. Au delà, cet énoncé propose un moyen de fusionner des fichiers de plusieurs millions de lignes.

## Correction

```
# coding: latin-1
# Question 1

def convertit_fichier_en_matrice (file):
    f = open (file, "r")
    l = f.readlines ()
```

```

f.close ()

mat = list()
for s in l :
    l = s.strip ("\n\r").split ("\t")
    mat.append (l)
return mat

# Question 2

def convertit_matrice_en_dictionnaire (matrice) :
    d={}
    for line in matrice :
        d[line[0],line[1]] = line [2:]
    return d

# Question 3

def fusion_dictionnaire (dico1, dico2) :
    dico={}
    for k in dico1 :
        if k in dico2 : dico[k] = dico1[k] + dico2[k]
    return dico

# Question 4

def convertit_dictionnaire_en_matrice (dico) :
    m =[]
    for k,v in dico.iteritems () :
        line = list (k) + v
        m.append (line)
    return m

# Question 5

def convertit_matrice_en_fichier (mat,nomfichier):
    f = open (nomfichier, "w")
    for line in mat :
        f.write ( "\t".join ( [ str (x) for x in line ] ) + "\n")
    f.close ()

# Question 6

def fusion_fichier (fichier1, fichier2, fichier_resultat) :
    matrice1 = convertit_fichier_en_matrice(fichier1)
    matrice2 = convertit_fichier_en_matrice(fichier2)
    dico1 = convertit_matrice_en_dictionnaire(matrice1)
    dico2 = convertit_matrice_en_dictionnaire(matrice2)
    dico = fusion_dictionnaire (dico1,dico2)
    matrice = convertit_dictionnaire_en_matrice(dico)
    convertit_matrice_en_fichier (matrice,fichier_resultat)

fusion_fichier ( "td_note_2009_cluedo_1.txt",
                 "td_note_2009_cluedo_2.txt",
                 "cluedo.txt")

# Question 7

def fusion_dictionnairebis (dico1,dico2) :

```

```

l1=dico1.keys()
l2=dico2.keys()
dico={}
for k in l1 :
    if k not in l2 : dico[k]=[]
for k in l2 :
    if k not in l1 : dico[k]=[]
return dico

def union_moins_intersection_fichier (fichier1, fichier2, fichier_resultat):
    matrice1 = convertit_fichier_en_matrice(fichier1)
    matrice2 = convertit_fichier_en_matrice(fichier2)
    dico1 = convertit_matrice_en_dictionnaire(matrice1)
    dico2 = convertit_matrice_en_dictionnaire(matrice2)

    dico = fusion_dictionnairebis (dico1,dico2)

    matrice = convertit_dictionnaire_en_matrice(dico)
    convertit_matrice_en_fichier (matrice,fichier_resultat)

union_moins_intersection_fichier ( "td_note_2009_cluedo_1.txt",
                                   "td_note_2009_cluedo_2.txt",
                                   "cluedo2.txt")

# Question 8
"""
Il suffit de fusionner les fichiers deux par deux
en procédant par récurrence. On fusionne d'abord les
deux premiers, puis on fusionne le troisième
au résultat de la première fusion...
"""

```