

---

## TP 7, Entrées/sorties : Corrigé.

---

### 2 Exercices de mise en pratique

**Exercice 1.** *Fichier contenant des entiers.* On ouvre le fichier, on lit chaque ligne en convertissant en entier, et on fait la somme. Pas besoin de supprimer le saut de ligne, `int("42\n")` s'évalue bien en 42, par exemple.

```
f=open('liste_entiers.txt','r')
s=0
for x in f:
    s+=int(x)
print(s)
```

**Exercice 2.** *Fichier contenant des flottants.* Avec `matplotlib.pyplot` importé :

```
g=open('liste_couples.txt','r')
X,Y=[],[]
for x in g:
    t=x[:-1].split(';')
    X.append(float(t[0]))
    Y.append(float(t[1]))
plt.plot(X,Y)
plt.show()
```

Explication : chaque ligne est de la forme `"42.0;3.14\n"`. Utiliser `split` avec le caractère `";"` permet d'obtenir la liste `["42.0", "3.14\n"]`. On convertit les deux composantes en flottants, et on les stocke dans deux listes idoines. Une fois le fichier parcouru, il suffit de tracer.

**Exercice 3.** *Écriture dans un fichier.*

```
def creation(f,n):
    for i in range(1,n+1):
        f.write(str(i)+"\n")
    f.close()
```

### 3 Exercices plus conséquents

**Exercice 4.** *Génération des notes du trimestre.* On suppose `randint` importée.

1. 

```
def genere_note():
    a=randint(0,40)
    return a/2
```

2. On fait usage de concaténation de chaînes (via `+`) et de conversion en chaîne (via `str`). On garde les lignes de `f1` entières, sauf qu'on supprime le saut de ligne : avec `x` une ligne, `x[:-1]` est une copie de la ligne sans son dernier élément (précisément le saut de ligne!).

```
def genere_notes(f1,f2):
    for x in f1:
        n1,n2=genere_note(),genere_note()
        f2.write(x[:-1]+","+str(n1)+","+str(n2)+"\n")
```

**Exercice 5.** *Calcul de la moyenne.* Dans le même style :

1. 

```
def moyenne(n1,n2):
    return (n1+n2)/2
```

2. 

```
def moyennes(f1,f2):
    for x in f1:
        n,p,n1,n2=x[:-1].split(',')
        f2.write(",".join([n,p,str(moyenne(float(n1),float(n2))]))+"\n")
```

**Exercice 6.** *La moyenne de la classe.* C'est très semblable à l'exercice 1. Pour connaître le nombre d'élèves de la classe, on peut ajouter un compteur, ou tout simplement utiliser `readlines` : on obtient une liste de lignes. Chaque ligne `x` étant de la forme `"nom,prenom,note\n"`, on peut utiliser `split` avec `,`, et récupérer simplement l'élément d'indice 2 de la liste obtenue (la note), qu'on convertit en flottant.

```
def moyenne_classe(f):
    s=0
    T=f.readlines()
    for x in T:
        s+=float(x[:-1].split(',')[2])
    print("Moyenne de classe : ",s/len(T))
```

**Exercice 7.** *Rang.*

```
def rang(f1,f2):
    T=[x[:-1].split(',') for x in f1.readlines()]
    T.sort(key=lambda x:20-float(x[2]))
    n=len(T)
    for i in range(n):
        T[i].append(str(i+1))
    T.sort()
    for x in T:
        f2.write(",".join(x)+"\n")
```

Explications :

- le fichier contient des lignes de la forme `"nom,prenom,note\n"`.
- la première ligne extrait toutes les lignes du fichier, et exécute `split` dessus après suppression du saut de ligne, à l'aide de la virgule. Les éléments de `T` sont donc de la forme `["nom", "prenom", "note"]`.
- La deuxième ligne trie `T` en place, par valeur de note décroissante.
- La boucle `for` rajoute le rang (converti en chaîne) à chaque élément de `T`.
- La ligne qui suit trie `T` à nouveau. Ici aucune fonction n'est précisée, et le tri se fait par défaut : ici cela sera par ordre alphabétique du nom (première composante de chaque élément de `T`), ce qui nous convient !
- Enfin, on réécrit les éléments de `T` dans `f2`, avec un appel à `join`.