TP d'informatique n°08

Manipulations de fichiers

L'objectif du TP est d'apprendre à extraire des données d'un fichier txt ou csv par un programme Python, apprendre à écrire des données simples dans un fichier texte.

0) Rappel des fonctionnalités

Tout d'abord une remarque : IDLE exécute le programme dans le dossier où est enregistré le fichier édité mais cela n'est pas le cas de Pyzo. Donc on commencera par effectuer un changement de répertoire courant à l'aide des commandes du type : (évidemment vous mettrez votre propre dossier)

from **os** import **chdir**

chdir ("C:\\Users\\François\\Documents\\calio\\mpsi20142015\\Informatique\\")

Commandes usuelles

- Ouvrir le fichier appelé 'fichier.txt' en définissant l'objet-fichier **f** :
  f = open('fichier.txt' , 'r')
- Créer le fichier appelé 'fichier.txt' en définissant l'objet-fichier **f** :
  f = open('fichier.txt' , 'w')
- Compléter le fichier appelé 'fichier.txt' en définissant l'objet-fichier **f** :
  f = open('fichier.txt' , 'a')
- Ecrire dans le fichier géré par l'objet-fichier **f** :
  f .write('mot à ajouter')
- Ecrire dans le fichier géré par l'objet-fichier **f** et passer à la ligne:
  f .write('mot à ajouter\n') ou **f .write('mot à ajouter' + '\n')**
- Lire une ligne dans le fichier géré par l'objet-fichier **f** et passer à la ligne:
  f .readline()
- Fermer le fichier géré par l'objet-fichier **f**:
  f .close()

On a les mêmes fonctionnalités pour les fichiers .csv
Le module csv contient d'autres fonctionnalités pour ce type de fichier

Recopier dans votre répertoire de travail les fichiers nécessaires au TP : decimales.txt, naissances2012.csv et valeurs.txt

1) Exercice 1

1) **C**réer un fichier texte appelé 'table7.txt' qui écrit la table de multiplication de 7 à l'aide d'une boucle **for**
   Ce fichier contiendra 10 lignes. Par exemple la deuxième ligne sera :
   2 * 7 = 14
   Rappel : on convertit un entier en une chaine de caractères à l'aide de la fonction **str**
2) **O**uvrir votre fichier créé avec le Bloc-Note de Windows (ou Wordpad) et vérifier qu'il a été correctement rempli

2) Exercice 2

1) **O**uvrir (avec Bloc-Note) le fichier texte appelé 'valeurs.txt', dont chaque ligne est la représentation d'une valeur numérique de type flottant. Par exemple :

   14.896
   7894.12
   12.2415
   etc …

2) **E**crire un script Python qui recopie ces valeurs dans un autre fichier appelé 'arrondis.txt' en arrondissant chaque valeur à l'entier le plus proche
3) **O**uvrir votre fichier créé avec le Bloc-Note de Windows (ou Wordpad) et vérifier qu'il a été correctement rempli

3) Exercice 3

Un fichier CSV est un fichier texte, lisible par un tableur (OpenOffice, Excel, …), contenant des données sur chaque ligne séparés par un caractère de séparation (une virgule, un point-virgule, une tabulation,…). Par exemple, les premières lignes du fichier 'naissances2012.csv' sont :
Achères, 4
Ainay-le-Vieil, 3
Les Aix-d'Angillon, 13

Ce fichier contient le nombre de naissances d'enfants domiciliés dans chacune des communes de la région Centre. Le caractère de séparation choisi est ici la virgule. Les communes sont classées par ordre alphabétique par département. Ces données proviennent du site de l'INSEE.

1) **O**uvrir le fichier 'naissances2012.csv' avec un tableur
2) **O**uvrir le fichier 'naissances2012.csv' en Python et écrire un script qui compte le nombre de communes dans la région Centre (c'est-à-dire le nombre de lignes du fichier).

3) **M**odifier ce script pour créer deux listes communes et naissances. La liste communes sera une liste de chaines de caractères. Ainsi communes = ['Achères', 'Ainay-Le-Vieil', 'Les Aix-d'Angillon', ….
La liste naissances sera une liste d'entiers. Ainsi naissances = [4, 3, 13, …
*Aide : utiliser la méthode **split** qui permet de séparer une chaine de caractères selon un caractère donné en paramètre. Par exemple,*

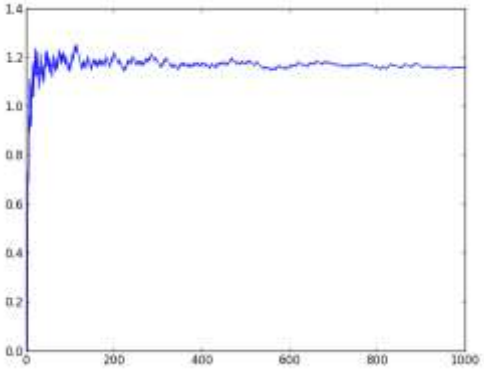
```
>>> s = 'Achères,4'
>>> s.split(',')
['Achères', '4']
```

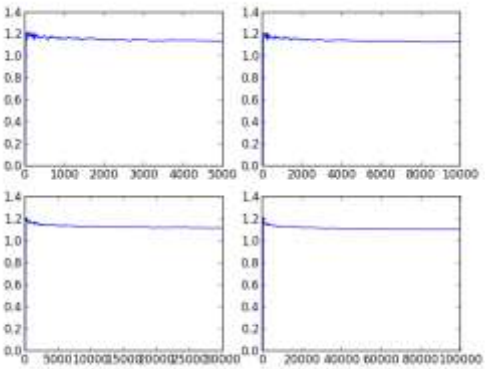
4) **E**crire un script qui calcule le nombre total de naissances dans la région Centre.
5) **E**crire un script qui affiche le nom des communes ayant eu plus de 100 naissances en 2012.
6) **M**odifier ce script pour créer un fichier texte contenant le nom des communes ayant eu plus de 100 naissances en 2012

4) Exercice 4

1) **E**crire dans un fichier que vous nommerez 'nb_premiers.txt' les nombres premiers inférieurs ou égaux à 10000.
*On pourra soit écrire une fonction **premier** qui renvoie **True** ou **False** selon que l'entier soit premier ou non, soit utiliser le crible d'Eratosthène.*

2) **L**ire ensuite ce fichier, et afficher la somme des nombres premiers inférieurs à 10000
3) **E**crire une fonction **compte** qui prend en entrée un entier N avec $1 \leqslant N \leqslant 10000$, et qui compte le nombre de nombres premiers compris entre 1 et N en utilisant le fichier 'nb_premiers.txt'. Tester votre **compte** pour N = 100, 500, 1000, 5000 puis 10000.
4) **V**érifier sur ces exemples le *théorème des nombres premiers* qui stipule que si $\pi(x)$ est le nombre de nombres premiers inférieurs à x, alors $\pi(x) \sim \frac{x}{\ln(x)}$
On observera graphiquement que $\frac{\pi(x)\ln(x)}{x}$ *tend vers* 1 *quand x tend vers* $+\infty$

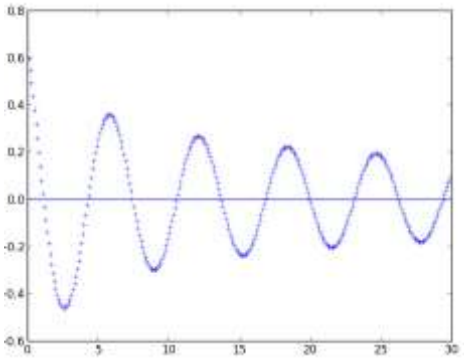




5) Exercice 5

On enregistre via LatisPro des données correspondant à la décharge d'un condensateur dans un circuit RLC en régime libre. On récupère ces données dans un fichier appelé RLC.txt dont les premières données sont indiquées dans la partie gauche de la figure ci-dessous. Le graphe correspondant au signal est donné à droite (en abscisses : le temps t en secondes et en ordonnées, la tension en Volts)

```
Temps;EA0
0;0.039140762761235
5E-6;0.0840135318040848
1E-5;0.10135318040848
1.5E-5;0.14112987332046
2E-5;0.18112987332046
2.5E-5;0.222124428600073
3E-5;0.2621706239879
3.5E-5;0.323118983879685
```



1) **L**e fichier RLC.txt peut contenir plusieurs milliers de lignes, il convient donc d'automatiser la récupération des données. Ecrire une procédure **lecture(fichier)** prenant comme argument une chaine de caractères **fichier** qui donne l'adresse où se trouve le fichier sur l'ordinateur et renvoie deux listes : respectivement la liste des temps et la liste des tensions. On veillera à ne pas prendre en compte toute ligne commençant par **Temps**, séparera les colonnes selon le caractère ; (à l'aide de la méthode **split(';')** et convertira les chaines de caractères correspondant au temps et à la tension en flottants.

2) Les deux listes étant données, on essaie de trouver la pseudo-période. On commence par écrire une fonction **maxima_successifs(tensions)** qui prend comme argument une liste de flottants et qui renvoie une liste des positions **i** de ces maxima. On considère que le premier maximum ne peut avoir lieu qu'après al première annulation et qu'on aura forcément qu'un seul maximum entre la première et la troisième annulation, puis un autre entre la troisième et la cinquième, etc …

3) On essaie de trouver une estimation de la pseudo-période

a) Une première idée est de considérer que cette pseudo-période est le double du temps séparant deux passages successifs par 0. Ecrivez une fonction **pseudo_periode_par_zeros(temps, tensions)** qui prend en argument une liste (triée) des temps et une liste des tensions associées à ces temps et renvoie une estimation de la pseudo-période, par exemple en donnant une moyenne des 10 premières pseudo-périodes.

b) La pseudo-période T est aussi égale (approximativement) au temps séparant deux maximas consécutifs. En regardant à nouveau au maximum les dix premières pseudo-périodes pour éviter les éventuelles erreurs de mesures sur les petites tensions en fin d'amortissement, écrire une fonction **pseudo_periode_par_maxima(temps, tensions)** qui prend en argument une liste (triée) des temps et une liste des tensions associées à ces temps et renvoie une estimation de la pseudo-période.