

<p style="text-align: center;"><b>Informatique en CPGE (2018-2019)</b> <b>TP 7 : les fichiers</b></p>
---

Les fichiers textes seront créés dans le dossier où sont enregistrés les fichiers Python.

### Exercice 1

1. Ecrire un programme qui crée et ouvre en écriture un fichier nommé "exercice1.txt", dans lequel il écrit un entier, un réel, un caractère et un mot, séparés par une tabulation. Le fichier est ensuite fermé.
2. Ecrire un programme qui ouvre le fichier "exercice1.txt" en lecture.  
Ecrire les commandes qui suivent et observer le résultat.

```
ligne=fic.readline()
liste=ligne.split('\t')
print(len(ligne))
print(ligne)
print(len(liste))
print(liste)
```

Compléter le programme afin qu'il affiche la somme des deux nombres puis le caractère et le mot séparés par une virgule.

Le fichier est ensuite fermé.

3. Ecrire un programme qui ouvre à nouveau le fichier "exercice1.txt" en écriture et ajoute un mot sur une nouvelle ligne. Vérifier que cette dernière étape s'est bien déroulée.

### Exercice 2

1. Ecrire un programme qui crée un fichier "espace.dat" dans lequel il écrit sur chaque ligne les coordonnées  $x, y, z$  d'un point de l'espace, séparées par une tabulation. On donne  $z = x^2 - y^2$  et on fera varier chacune des valeurs  $x$  et  $y$  dans l'intervalle  $[-5; 5]$  avec un pas de 0,1.
2. Ouvrir le logiciel **Gnuplot** et écrire la commande **splot 'adresse du fichier ...espace.dat'** pour tracer les points. On peut aussi aller dans le menu **3D Plot**, puis sélectionner **Data Filename ...** et chercher le fichier dans l'arborescence.

### Exercice 3

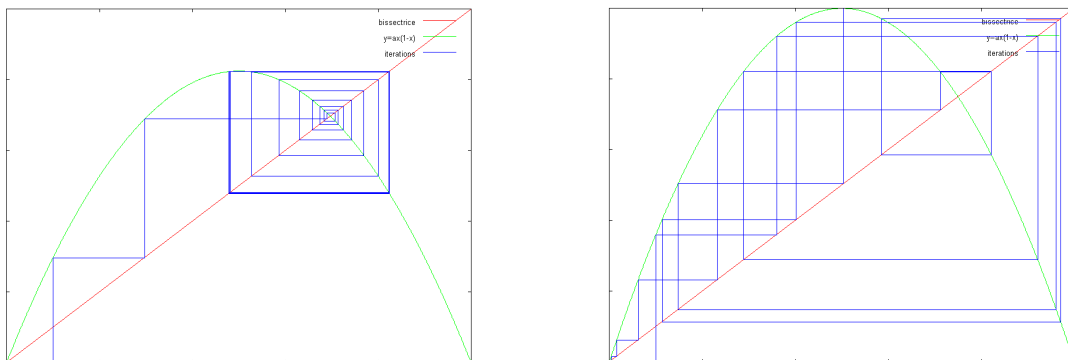
1. Ecrire un programme qui demande à l'utilisateur d'entrer le nom d'un fichier avec l'extension ".dat", (par exemple "marche.dat"), dans le but de le créer, de l'ouvrir puis d'y écrire. Vérifier que cette étape s'est déroulée avec succès.
2. Compléter le programme afin qu'il écrive dans le fichier 101 lignes où sur chaque ligne seront écrites l'abscisse et l'ordonnée d'un point, séparées par une tabulation. Pour la première ligne, l'abscisse et l'ordonnée seront égales à 0. Pour les lignes suivantes l'abscisse sera entière et prendra les valeurs 1, 2, 3, ..., 99, 100. Pour chaque abscisse, l'ordonnée correspondante sera la somme de l'ordonnée précédente et d'un entier aléatoire compris entre  $-5$  et  $5$ .
3. Exécuter le programme et vérifier le contenu du fichier créé.
4. Ouvrir le logiciel **Gnuplot** et écrire la commande **plot 'adresse du fichier ...marche.dat' w l** pour tracer les points et les lier ( $w l =$  with lines). On peut aussi aller dans le menu **Plot**, puis sélectionner **Data Filename ...** et chercher le fichier dans l'arborescence.
5. Compléter le programme afin d'ouvrir en lecture le fichier créé et récupérer dans deux listes **xliste** et **yliste** les abscisses et les ordonnées écrites dans le fichier.

- Utiliser la bibliothèque **Matplotlib** pour tracer le graphique ; pour cela ajouter à la fin du programme les lignes suivantes :

```
import matplotlib.pyplot as plt
plt.plot(xliste,yliste)
plt.show()
```

#### Exercice 4

Il s'agit d'écrire un programme permettant d'obtenir des figures illustrant la construction graphique des termes d'une suite récurrente définie par  $u_0$  et  $u_{n+1} = f(u_n)$ . La fonction  $f$  est définie sur  $[0; 1]$  par  $f(x) = ax(1 - x)$  avec  $a \in ]0; 4]$  et  $u_0 \in ]0; 1[$ . On obtiendra en particulier le type d'images qui suit :



- Ecrire un programme qui crée un fichier nommé "bissectrice.dat" et qui écrit dans ce fichier les coordonnées des deux points extrêmes de la fonction  $y = x$  avec  $x \in [0; 1]$ . Chaque ligne du fichier contient l'abscisse et l'ordonnée d'un point séparées par une tabulation.
- Ouvrir le logiciel Gnuplot et vérifier le résultat obtenu. On écrit la commande **plot 'adresse du fichier ...bissectrice.dat' title 'bissectrice' w l**.
- Ecrire la définition de la fonction  $f$  qui prend en arguments  $x$  et  $a$  et renvoie la valeur de  $f(x)$ . Ensuite, le programme demande à l'utilisateur la valeur de  $a$ , puis crée un fichier nommé 'courbe.dat' où il écrit un tableau de valeurs de la fonction  $y = f(x)$  sur  $[0; 1]$  avec un pas de 0,001.
- Vérifier avec Gnuplot qu'on obtient bien le tracé de la bissectrice et de la courbe.
- Le programme demande à l'utilisateur la valeur de  $u_0$ , le nombre d'itérations "imax" et écrit dans un troisième fichier nommé "iter.dat" les coordonnées des points permettant le tracé des itérations. On commencera par écrire les coordonnées du premier point  $(u_0; 0)$  ; puis (à l'aide d'une boucle `for i in range (imax) :`), on écrira les coordonnées des points  $(x; y)$  et  $(y; y)$  avec  $y = f(x)$ .
- Tester le programme avec  $a = 3, 2$ ,  $u_0 = 0, 1$  et  $\text{imax} = 10$ , puis ouvrir Gnuplot pour obtenir la figure. Tester ensuite différentes valeurs pour  $u_0$  et  $a$ , par exemple  $a = 1$  avec  $x_0 = 0, 3$  ou  $0, 7$ , ou  $a = 1, 6$ ,  $a = 2, 8$ ,  $a = 3, 2$  et  $a = 4$ . Déterminer des valeurs de  $a$  qui correspondent à différents types de graphiques (escalier, spirale, cycle, chaos).

Note : on peut enregistrer les images créées par Gnuplot au format désiré, (eps, png, gif, jpg), avec les commandes **set terminal jpeg**, puis **set output 'monimage.jpg'**.