Licence Informatique — **INF158 – Projet de programmation 1**
Université Bordeaux 1 — **Feuille 2**

Placez vous dans votre répertoire correspondant au projet de programmation 1 et téléchargez les fichiers mis à votre disposition à l'aide de la commande :

wget http://dept-info.labri.fr/ENSEIGNEMENT/projetprog1/src/fichiers-2.tar.gz

De la documentation (bibliographie, liens, etc.) est disponible via la page de l'unité d'enseignement : http://dept-info.labri.fr/ENSEIGNEMENT/projetprog1/

Gestion des fichiers

Premières expérimentations

Étape 1 — *lecture, écriture ligne par ligne*

1. Écrire une fonction python `recopie_et_numerote_les_lignes(nom_fichier_entree, nom_fichier_sortie)` qui lit *ligne par ligne* le contenu du fichier texte `nom_fichier_entree` et copie dans un fichier texte `nom_fichier_sortie` chaque ligne du fichier d'entrée en rajoutant au début de chaque ligne le numéro de la ligne. Vous pourrez tester votre fonction en utilisant le fichier `texte-1.txt` comme fichier d'entrée.
2. Modifier cette fonction de façon à ignorer les lignes commençant par le caractère `#`.

Étape 2 — *conversion entier / chaîne de caractères*
Éditez le fichier `triangle-de-pascal.py`. Écrire le code python permettant de stocker les `NOMBRE_DE_LIGNES` premières lignes du triangle de Pascal dans un fichier texte.

Étape 3 — *concaténation*
Écrire une fonction `concatener(nom_fichier_1, nom_fichier_2)` qui ajoute à la fin du fichier `nom_fichier_1` le contenu du fichier `nom_fichier_2`.

Étape 4 — *lecture, écriture mot par mot*
Écrire une fonction `calcul_somme_des_lignes(nom_fichier_entree, nom_fichier_sortie)` qui lit en entrée un fichier `nom_fichier_entree` contenant des lignes de nombres entiers (sous forme de chaînes de caractères), et qui copie dans un fichier `nom_fichier_sortie` la somme des nombres de chaque ligne de `nom_fichier_entree`. Vous pourrez utiliser `liste-de-nombres.txt` comme fichier d'entrée.

Application aux fichiers images (format PGM)

Le format PGM (Portable Grey Map) fait partie d'un ensemble de six formats, génériquement appelés PBM, et codant en ASCII ou en binaire des images noir et blanc, monochromes ou couleurs. Nous allons utiliser le format PGM qui définit un codage ASCII des images en niveau de gris.

Rappel des caractéristiques d'un fichier PGM — La première ligne comporte une suite de deux caractères décrivant le format du fichier. Le format retenu ici est `P2` (image en niveaux de gris dont la valeur de chaque pixel est écrite en ASCII). La ligne suivante comporte la largeur et la hauteur de l'image, en ASCII. La troisième ligne comporte la valeur maximale du niveau de gris, également en ASCII. Cette valeur est généralement fixée à 255 (256 niveaux de gris différents). La quatrième ligne est constituée de la suite de tous les pixels composant l'image, ligne après ligne, chacun ayant une valeur entre 0 (noir) et 255 (blanc). Les codes de deux pixels consécutifs sont séparés par un espace. Cette quatrième ligne peut être décomposée en plusieurs lignes (par exemple une ligne de valeurs pour chaque ligne de l'image ou une ligne par valeur).

Étape 5 — *création d'un fichier PGM*
Écrire une fonction `fichier_zebrures(hauteur, largeur, taille_zebrure, nom_fichier)` qui crée un fichier texte codant une image PGM de taille `hauteur`×`largeur` et contenant des bandes horizontales (ou zébrures) alternées noires et blanches de hauteur `taille_zebrure` (sauf éventuellement pour la hauteur de la dernière zébrure).

Étape 6 — *création d'une image PGM en mémoire et sauvegarde dans un fichier*
Pour stocker en mémoire une image PGM, les niveaux de gris de chaque pixel seront stockés dans un tableau à une dimension ; la hauteur et la largeur de l'image seront stockées dans des variables `hauteur` et `largeur`.

1. Écrire une fonction `image_zebrures_horizontales(hauteur, largeur, taille_zebrure, niveau_gris_zebrure_1, niveau_gris_zebrure_2, image)` qui stocke dans un tableau `image` les niveaux de gris des pixels d'une image PGM, de taille `hauteur`×`largeur` et contenant des zébrures alternées de couleur `niveau_gris_zebrure_1` et `niveau_gris_zebrure_2`. La hauteur d'une zébrure est donnée par `taille_zebrure` (sauf éventuellement pour la dernière zébrure).
2. Écrire une fonction `sauvegarder_image_PGM_ascii(hauteur, largeur, image, nom_fichier_image)` qui sauvegarde dans un fichier `nom_fichier_image` l'image contenue dans le tableau `image`.

Étape 7 — *lecture d'un fichier PGM et stockage de l'image en mémoire*
Écrire une fonction `lire_fichier_PGM_ascii(nom_fichier, image)` qui lit le fichier `nom_fichier` (correspondant à une image PGM) et remplit le tableau `image` avec les niveaux de gris des pixels de l'image PGM. La fonction renverra le tableau `image` et affectera la taille de l'image aux variables globales `hauteur` et `largeur`.