

Dans cet exercice, nous allons lire et traiter des données qui sont stockées dans un fichier, et qui concernent les patients d'un cabinet médical. On suppose que ce fichier est de la même forme que les fichiers `donnees1.txt` et `echantillon-donnees.txt` (qui vous sont fournis, et sur lesquels vous pouvez faire vos tests), c'est-à-dire : chaque ligne correspond à un patient, et est composée du prénom du patient, suivi d'un espace, puis du poids du patient, suivi d'un espace, puis de la taille du patient. Le but global de cet exercice est de :

- lire les données dans le fichier, et les stocker sous la forme d'une liste de patients; chaque patient sera représenté par un dictionnaire, ayant trois clés: "nom", "poids" et "taille".
- calculer l'IMC (Indice de Masse Corporelle) de chaque patient, ainsi que l'IMC moyen des patients; détecter les patients qui sont en corpulence normale et ceux qui ne le sont pas.
- écrire les résultats dans un autre fichier.

Partie 1 : importer les données et les mettre dans la bonne structure de données

Ecrire une fonction `import_lignes` qui prend en argument un nom de fichier et qui renvoie la liste des lignes du fichier, c'est-à-dire une liste de chaînes de caractères, où chaque élément de la liste correspond à une ligne du fichier. Pensez bien à fermer le fichier avant de faire le return.

Ecrire une fonction `cree_patient` qui prend en argument une ligne (chaîne de caractères) et qui renvoie le dictionnaire correspondant au patient de la ligne. Pour cela, aidez-vous de la fonction `separe(ligne)` qui vous est fournie et renvoie une liste contenant trois chaînes de caractères, obtenue en séparant la ligne le long des espaces. Pensez à convertir le poids et la taille en float avant de les stocker dans le dictionnaire.

Exemples:

`separe("Liliane 62 1.67\n")` renvoie `['Liliane', '62', '1.67']`.

`cree_patient("Liliane 62 1.67\n")` doit renvoyer `{'nom' : 'Liliane', 'poids' : 62.0, 'taille': 1.67}`

Ecrire une fonction `liste_patients_from_nom_fichier` qui prend en argument un nom de fichier et qui renvoie la liste des patients correspondant au fichier (où chaque patient est représenté par un dictionnaire).

Partie 2: Traiter les données

Ecrire une fonction `imc` qui prend en argument un patient (dictionnaire), et qui renvoie son IMC arrondi à deux chiffres après la virgule (rappelez-vous l'existence et le fonctionnement de `round(nombre, 2)`). L'IMC se calcule ainsi: $IMC = \text{poids} / (\text{taille}^2)$.

Ecrire une fonction `imc_moyen` qui prend en argument une liste de patients et qui renvoie l'IMC moyen des patients, arrondi à deux chiffres après la virgule.

Ecrire une fonction `liste_noms_patients_en_corpulence_normale` qui prend en argument une liste de patients et qui renvoie la liste contenant les noms des patients dont l'IMC indique une corpulence normale, c'est-à-dire si l'IMC est compris entre 18.5 et 25 (inclus).

Partie 3: écrire les résultats dans un fichier

Attention: le fonctionnement de Caseine ne permet pas de conserver les modifications faites aux fichiers par le programme : tout est remis à zéro à la fin du Run ou du Evaluate. Pour tester cette partie, il vous faut donc, au choix:

Soit travailler sous Idle, puis uploader votre soumission

Soit utiliser la fonction `affiche_console(nom_fichier)` qui vous est fournie et qui vous permet d'afficher à la console le contenu du fichier dont le nom est passé en argument (voir l'exemple d'utilisation dans le programme principal).

Ecrire une fonction `produire_chaine` qui prend en argument un patient et qui renvoie la chaîne de caractères formée du nom du patient, suivi d'un espace, suivi de son IMC arrondi à deux chiffres après la virgule, suivi d'un retour à la ligne.

Ecrire une fonction `ecrire_imc` qui prend en argument la liste des patients ainsi que le nom du fichier de destination, et qui écrit dans le fichier:

Pour chaque patient, une ligne contenant son nom puis son IMC, séparés par un espace (pensez à utiliser la fonction `produire_chaine`),

Puis on écrit "IMC Moyen :" suivi de l'IMC moyen des patients, arrondi à deux chiffres après la virgule.

Ensuite, on écrit "Noms des patients en corpulence normale :", suivi d'un retour à la ligne, puis les noms des patients en corpulence normale (un nom par ligne)

Vous pouvez consulter le fichier `resultats-donnees1.txt` pour voir un exemple de résultat attendu (les données utilisées pour cet exemple sont issues du fichier `donnees1.txt`).

Ecrire une fonction `traitement_complet_donnees` qui prend en argument le nom du fichier contenant les données, puis le nom du fichier dans lequel on doit écrire les résultats, et qui effectue le traitement complet des données (c'est-à-dire qu'il lit, traite les données, et écrit les résultats selon les instructions données dans la question précédente). Vous pouvez consulter le fichier `resultats-donnees1.txt` pour voir ce que doit produire l'appel de `traitement_complet_donnees("donnees1.txt", "resultats-donnees1.txt")`

(Source pour la formule de calcul et l'interprétation de l'IMC:
<https://www.imc.fr/calcul>)