

DEVOIR D'INFORMATIQUE N° 2 (2 HEURES)

Ce devoir est constitué de plusieurs petits exercices. L'ordre des exercices ne correspond à aucun critère de difficulté ou de longueur : vous pouvez les traiter dans l'ordre que vous voulez. Veillez à soigner la copie tant pour l'écriture, la propreté que pour la rédaction, la rigueur et l'argumentation. De plus, on prêtera une attention particulière au respect des alignements et des indentations des séquence d'instructions Python. La calculatrice est autorisée.

Vous numéroterez vos copies et ferez apparaître clairement sur la première page le nombre de copies.

Exercice 1

1. Ecrire une fonction **moy_ecartype** prenant comme argument une liste **L** de nombres et qui retourne le couple (sous forme de tuple ou liste au choix) constitué de la moyenne et de l'écart-type de **L** (*écart-type* : racine de la moyenne des carrés des écarts à la moyenne). Si la liste est vide, la fonction peut retourner au choix le couple (0,0) ou le booléen 'False' ou la valeur sans type 'None'
2. Ecrire une fonction **moy_sup10** prenant comme argument une liste **L** de nombres et qui retourne la moyenne des ses termes strictement supérieurs à 10. Si aucun des termes n'est supérieur à 10, le retour doit être le nombre 0.
3. Ecrire une fonction **moy_indeximpair** prenant comme argument une liste **L** de nombres et qui retourne la moyenne de ses termes d'indice impair.
4. Ecrire une fonction **moy_pair** prenant comme argument une liste **L** d'entiers et qui retourne la moyenne de ses termes pairs.

Exercice 2

On rappelle qu'une chaîne de caractères n'est pas "modifiable" : on ne peut pas faire une affectation simple changeant un caractère donné : il faut recréer une nouvelle variable constituant la chaîne de caractères souhaitée.

1. Ecrire une séquence d'instructions permettant de créer 3 listes :
 - la liste **majuscules** contenant toutes les lettres majuscules (la liste commençant donc par ['A', 'B', 'C'])
 - la liste **minuscules** contenant toutes les lettres minuscules non accentuées (la liste commençant donc par ['a', 'b', 'c'])
 - la liste **alphabet** contenant toutes les lettres non accentuées et sans attributs de l'alphabet français usuel

Pour la suite de l'exercice, on pourra utiliser les trois listes précédentes sans en donner le principe de construction

2. Ecrire une fonction **enleve_e_aigu** prenant comme argument une chaîne de caractères **phrase** et qui retourne la chaîne de caractères obtenue à partir de **phrase** en changeant tous les **é** en **e**.
3. Ecrire une fonction **En_minuscule** prenant comme argument une chaîne de caractères **phrase** et qui retourne la chaîne de caractères obtenue à partir de **phrase** en changeant toutes les majuscules en la minuscule correspondante.
4. Ecrire une fonction **Enleve_accent_et_cedille** prenant comme argument une chaîne de caractères **phrase** et qui retourne la chaîne de caractères obtenue à partir de **phrase** en enlevant tous les accents et les cédilles (On supposera que les accents et cédilles ne sont portés que par les minuscules et qu'on utilise un texte français dont les seules lettres ayant ces attributs sont : à, â, ç, é, è, ê, ë, î, ï, ô, ü, ù, ÿ)
5. Ecrire une fonction **Nombre_e** prenant comme argument une chaîne de caractères **phrase** et qui retourne le nombre de **e** quelquesoit sa forme (accentuée ou non, majuscule ou non)

6. Ecrire une fonction **Frequence_voyelle** prenant comme argument une chaîne de caractères **phrase** et qui retourne la liste des 6 fréquences des voyelles **a, e, i, o, u, y** dans l'ordre, quelque soit leur forme (accentuées ou non, majuscules ou non) parmi les lettres de la chaîne de caractères **phrase**. Attention la fréquence de **e** dans la chaîne de caractères "Elles étaient 9, voire plus." est de 5/21 et non 5/28 : les espaces, chiffres, signes de ponctuation ne doivent pas être comptés comme des lettres.

Exercice 3 : Codage CLE

Soit N un entier naturel. On rappelle que N admet une représentation binaire sous forme d'une suite finie $(a_k)_{0 \leq k \leq p-1}$ de 0 et de 1, dont le dernier terme a_{p-1} est non nul.

Plus précisément, on aura : $N = a_0 + a_1 2^1 + a_2 2^2 + \dots + a_k 2^k + \dots + a_{p-1} 2^{p-1}$. L'entier p s'appelle la taille de N . Le nombre N est alors représenté par la liste $[a_0, a_1, \dots, a_{p-1}]$. L'entier 0 est représenté par la liste vide.

Dans le but d'écrire de manière courte les grands entiers, le codage CLE (Code Large Echelle) associe à l'entier N la liste L constituée des seules k tels que a_k vaut 1 dans la décomposition en base 2 de N .

Par exemple, 67 qui s'écrit $[1, 1, 0, 0, 0, 0, 1]$ en base 2, s'écrit $[0, 1, 6]$ en code CLE.

Questions mathématiques

- Calculer le nombre de chiffres nécessaires pour décrire en base 2, tous les entiers compris entre 0 et 15 (par exemple, les nombres compris entre 0 et 4 sont en base 2 représentés par : [], [1], [0, 1], [1, 1] et [0, 0, 1] ce qui nécessite 8 chiffres pour décrire en base 2 tous les entiers compris entre 0 et 4)
Calculer de même le nombre de chiffres nécessaires pour décrire en code CLE tous les entiers entre 0 et 15 (par exemple, les nombres compris entre 0 et 4 sont représentés en codage CLE par : [], [0], [1], [0, 1] et [2] ce qui nécessite 5 chiffres)
- On appelle M_p la moyenne du nombre de chiffres nécessaires pour décrire en base 2 tous les entiers compris entre 0 et $2^p - 1$ et C_p la moyenne du nombre de chiffres nécessaires pour décrire en code CLE tous les entiers compris entre 0 et $2^p - 1$ (c'est-à-dire, le nombre de chiffres utilisés divisé par 2^p).
 - Exprimer M_{p+1} en fonction de M_p et de p .
 - Exprimer C_{p+1} en fonction de C_p et de p .
 - En déduire les expressions de M_p et de C_p en fonction de p (on trouvera des expressions du type : $a + bp + c2^{-p}$ avec a, b et c des constantes), puis comparer des deux nombres, en particulier pour p tendant vers $+\infty$

Questions informatiques

- Ecrire une fonction **convbin** qui convertit un entier N (donné en base 10) en binaire
- Ecrire une fonction **convCLE** qui convertit un entier N (donné en base 10) en code CLE.
- Ecrire l'algorithme réciproque **deconvCLE** qui convertit une liste L correspondant au codage CLE d'un entier N en cet entier N
- Ecrire une procédure **divparpuis2** qui prend comme argument une liste L correspondant au codage CLE d'un entier N et un entier k et qui renvoie **True** si N est divisible par 2^k et **False** sinon. On demande de travailler directement sur la liste L et non de transformer cette liste en l'entier N correspondant

Questions mathématiques

En fait la présentation du codage CLE est incomplète : lorsque l'on représente 67 par $[0, 1, 6]$ en code CLE, on doit encore transformer ces chiffres 0, 1 et 6 en binaire. Et en fait on représente 67 par $[[0], [1], [0, 1, 1]]$ ce qui prend 5 bits.

- Combien faut-il de bits pour représenter en binaire le nombre 2^{1024} ? Combien en faut-il pour représenter ce nombre en codage CLE ?
- Mêmes questions avec le nombre $2^{1024} - 1$?