

1. Distance de Hamming

Créer la fonction *hamming*(mot_1, mot_2) qui calcule la « distance de Hamming » entre deux mots :

- Si les deux mots ont la même longueur, la distance est égale au nombre de positions pour lesquelles les deux mots ont des caractères différents.
- Si les deux mots n'ont pas la même longueur, renvoyer la valeur -1.

Exemples : *hamming*("ordinateur", "ordonneras") renvoie 6
car ces deux mots ont 6 différences : "ord*n*****"
hamming("informatique", "information") renvoie -1

2. Anagrammes

a) Créer la fonction *suppr*(mot, lettre) qui renvoie une chaîne de caractère correspondant au mot auquel on a supprimé la première apparition de la lettre donnée.

Remarque : Si la lettre n'est pas comprise dans le mot, la fonction renvoie alors le mot inchangé !

Exemple : *suppr*("octet", "t") renvoie "ocet"

b) Créer la fonction *scrabble*(mot, lettres) qui renvoie un booléen

- *True*, si on peut écrire le mot à l'aide des lettres sans utiliser deux fois la même lettre
- *False*, sinon

Exemples : *scrabble*("binaire", "aeiobcdnr") renvoie *True*
scrabble("positif", "aoipst") renvoie *False*

Indication : Utiliser la fonction *suppr*() du a)

c) Créer la fonction *anagramme*(mot_1, mot_2) qui renvoie un booléen

- *True*, si les mots sont des anagrammes l'un de l'autre
- *False*, sinon

Exemples : *anagramme*("tortue", "tourte") renvoie *True*
anagramme("touche", "mouche") renvoie *False*

Indication : Utiliser la fonction *scrabble*() du b)

3. Créer la fonction *jeu_de_cartes*(valeurs, couleurs) qui renvoie la liste de toutes les cartes d'un jeu de 52 cartes à l'aide des deux listes vues dans le TP_08 sur les fonctions :

valeurs = (2, 3, 4, 5, 6, 7, 8, 9, 10, "Valet", "Dame", "Roi", "As")
couleurs = ("pique", "cœur", "carreau", "trèfle")

4. Créer la fonction *nettoyer*(liste) qui supprime les doublons dans une liste.

Exemple : *nettoyer*([1, 2, "a", 1, 3, "a", 1]) renvoie [1, 2, "a", 3]

5. Créer la fonction *moyenne*(notes, coefficients) qui calcule la moyenne des notes avec leurs coefficients respectifs (par défaut ceux-ci pourront tous être égaux à 1) comme celle vue dans le TP_08 sur les fonctions, mais avec un nombre quelconque de notes dans une liste.

6. Créer la fonction *nb_entre*(liste, min, max) qui renvoie le nombre d'objets de la liste compris entre les bornes min et max (bornes comprises)

Exemple : *nb_entre*([1, 2, "a", 5, 3, "b", 2], 1, 3) renvoie 4

Attention : Utiliser *try ... except* pour gérer les erreurs dues à des objets de types différents.

7. Distance à une liste

- Pour les nombres, on utilisera la valeur absolue comme distance.
- Pour les chaînes de caractères, on utilisera la distance de Hamming vue dans le premier exercice.

a) Créer la fonction *distance*(liste, objet) qui renvoie la plus courte distance entre un élément de la liste et l'objet cité.

Exemple : *distance*([1, 2, 5, 9, 7, 5, 3], 6) renvoie 1

b) Créer la fonction *proches*(liste, objet) qui renvoie la liste des éléments de la liste, les plus proches de l'objet cité.

Exemple : *proches*([1, 2, 5, 9, 7, 5, 3], 6) renvoie [5, 7, 5]

8. Le numéro de sécurité social à 15 chiffres figurant sur la carte vitale comprend de nombreuses informations, en particulier :
On ne considérera que le cas le plus simple où :
- Le premier chiffre indique le sexe (1 : homme / 2 : femme)
 - Les deux suivants indiquent les deux derniers chiffres de l'année de naissance.
 - Les deux suivants indiquent le mois de naissance.
 - Les deux suivants indiquent le département de naissance.
 - Les deux derniers chiffres servent de clef de contrôle : C'est le complément à 97 du reste obtenu en prenant : le nombre formé par les 13 premiers chiffres et en le divisant par 97.

Exemple : Voici mon numéro de sécu : "168049202606447"

- "1" : Je suis un homme !
 - "68" : Je suis né en 1968
 - "04" : Je suis né en avril
 - "92" : Je suis né dans le département 92
 - "47" : $1680492026064 = 17324660062 \times 97 + 50$
et $97 - 50 = 47$ (utiliser le symbole % pour obtenir le reste)
- a) Créer la fonction *verif*(num) qui renvoie un booléen :
- *True*, si la clef de contrôle est correcte
 - *False*, sinon
- b) Créer la fonction *annee*(num) qui renvoie l'année de naissance (en supposant qu'il n'y a pas de centenaire !)
- c) Créer la fonction *mois*(num) qui renvoie le nom du mois de naissance (et non pas juste son numéro)
- d) Créer la fonction *politesse*(num) qui renvoie la phrase d'accueil : "Bonjour *monsieur/madame*, vous êtes né(e) en *janvier/.../décembre 1920/.../2019* dans le département *01/.../95*"

Exemple : "Bonjour monsieur, vous êtes né en avril 1968 dans le département 92"

Compléments sur la cryptographie

➤ Cryptage « César »

Le cryptage « César » est une méthode de cryptage de texte par décalage de chaque lettre d'une même distance dans l'alphabet. Il suffit de connaître la distance de décalage pour décoder le texte.

Exemple : Avec un décalage de 8.

Texte à coder : BONJOURMONSIEURBOND

Texte codé : JWVRWCZUWVAQMCZJWVVL

➤ Cryptage « Vigenère »

Le cryptage « Vigenère » est une méthode de cryptage de texte qui reprend le principe de décalage de chaque lettre mais la distance de décalage dépend d'un mot-clef choisi.

On répète ce mot-clef sous le texte original autant de fois que nécessaire, et chaque lettre du mot-clef indique par quelle lettre est codé le A dans le décalage pour la lettre du texte située juste au-dessus.

Exemple : Avec le mot-clef "CLEF".

Texte à coder : BONJOURMONSIEURBOND

Clef répétée : CLEFCLEFCLEFCLEFCLE

Texte codé : DZROQFVRQYWNGFVVGQYH