

■ T.P. 5 ■

Chiffre de César

On propose de chiffrer et déchiffrer un texte en utilisant un chiffre de type César. Le principe de ce chiffre est de décaler l'alphabet de k places en remplaçant la lettre numéro n par la lettre numéro $n + k$ modulo 26, la clé k étant un entier compris entre 0 et 25. Les fichiers à déchiffrer, `tp5_mystere1.txt` et `tp5_mystere2.txt`, sont à copier dans votre dossier de travail. On suppose que les caractères utilisés sont uniquement les minuscules de l'alphabet latin.

Partie I : Chiffrement / Déchiffrement élémentaire

1. Écrire une fonction `translate(liste,k)` qui, étant donnée une liste d'entiers `liste` compris entre 0 et 25 et un entier `k` renvoie la liste contenant les entiers de `liste` translatés de `k` modulo 26.
2. Écrire une fonction `occurrence(liste)` qui, étant donnée une liste `liste` d'entiers compris entre 0 et 25, renvoie une liste `occ` de taille 26 telle que `occ[i]` représente le nombre d'occurrences de l'entier `i` dans la liste `liste`.

Pour déterminer la clé d'un chiffre de César, il existe plusieurs méthodes :

- Essayer toutes les valeurs possibles de la clé puis laisser un utilisateur choisir la clé la plus vraisemblable.
 - Déterminer les occurrences de chacune des lettres, déterminer celle qui apparaît le plus souvent et supposer qu'il s'agit du code de la lettre `e` (lettre la plus fréquente en français, et cinquième lettre de l'alphabet).
3. Écrire une fonction `dechiffre_liste(liste)` qui, étant donnée une liste d'entiers `liste` représentant un texte, détermine l'entier le plus fréquent, en déduit la clé (supposée) `k` du chiffrement, puis renvoie le texte déchiffré (sous forme de liste d'entiers).

Partie II : Chaînes de caractères

En Python, les chaînes de caractères sont de type `str`.

- Les chaînes de caractère sont représentées entre guillemets, par exemple `'abc'` ou `"aujourd'hui"`.
- La chaîne vide est `' '`.
- Les indices et l'accès aux caractères sont identiques aux opérations sur les listes.
- Les éléments d'une chaîne ne sont pas mutables.

La commande `ord` permet de transformer un caractère en un entier (son code Utf-8) et la commande `chr` permet de transformer un entier en le caractère correspondant. On supposera dans toute la suite que le texte à chiffrer / déchiffrer est constitué uniquement de lettres minuscules non accentuées.

4. Écrire une fonction `char_to_int(caractere)` qui transforme un caractère (en minuscule) en un entier compris entre 0 et 25 (de façon bijective...).
5. Écrire une fonction `str_to_list(chaine)` qui transforme une chaîne de caractères en la liste d'entiers correspondante.
6. Écrire une fonction `list_to_str(liste)` qui effectue la transformation inverse.
7. Écrire deux fonctions `chiffre(texte,k)` et `dechiffre_occ(texte)` qui chiffrent et déchiffrant un texte en utilisant le chiffre de César. Pour le déchiffrement, on utilisera la méthode de la plus grande occurrence.
8. Tester votre fonction sur les textes contenus dans les fichiers mystères.

Partie III : Analyse de fréquences

La division n/p de deux entiers permet d'obtenir un *nombre flottant* (valeur approchée du réel n/p). Une deuxième méthode pour déterminer la clé `k` du chiffre de César est la suivante. On calcule, pour chaque valeur de `k`, la fréquence d'apparition de chacune des lettres dans le message chiffré. À l'aide d'un texte de référence, on détermine la fréquence d'apparition de chacune des lettres en français. On choisit la clé qui permet d'obtenir une liste de fréquences aussi proche que possible de la liste de référence. Pour cela,

si $f = (f_0, \dots, f_{25})$ et $g = (g_0, \dots, g_{25})$ sont deux listes de fréquences, on détermine leur distance via la fonction

$$d(f, g) = \sum_{j=0}^{25} |f_j - g_j|.$$

- 9.** Écrire une fonction `frequences(texte)` qui renvoie une liste `freq` telle que `freq[i]` soit la fréquence de la i ème lettre de l'alphabet dans le texte `texte`.
- 10.** Écrire une fonction `freq(texte,k)` qui renvoie la liste des fréquences des lettres du texte `texte` déchiffré grâce à la clé `k`.
- 11.** Écrire une fonction `distance(f,g)` qui renvoie la distance entre les listes de fréquences `f` et `g`.
- 12.** Écrire une fonction `dechiffre_freq(texte,freq_ref)` qui déchiffre le texte `texte` en utilisant la méthode des fréquences ainsi que la liste de fréquences de référence `freq_ref`.
- 13.** Tester votre fonction sur les fichiers mystères en utilisant le texte de référence disponible dans le fichier `tp5_reference.txt`.