Lycée Thiers — Informatique
Année 2015/16 — MPSI-PCSI

Feuille d'Exercices 9

Fichier crypté

Exercice 1. Copier le fichier `TexteCrypte.txt` sur votre ordinateur. Le sauvegarder dans le répertoire utilisateur (celui portant votre nom de login). Il s'agit d'un fichier texte contenant un texte célèbre qui a été crypté à l'aide d'un chiffrement de César (dont on ne connait pas la clé).

On souhaite le décrypter, le lire, et sauvegarder le texte déchiffré dans un fichier texte.

(1) Ouvrir le fichier `TexteCrypte.txt` en mode d'accès lecture (`'r'`).
(2) Récupérer dans une variable `texte` son contenu.
(3) Refermer le fichier. Afficher le contenu de `texte`.

Pour ce TD, on rappelle que les caractères accentués, chiffres et autres symbôles accessibles au clavier ou non, sont disponibles dans la table de caractères ASCII, numérotés de 0 à 255 (0xFF).
La fonction `chr(n)` retourne le caractère de la table de code ASCII `n`.
La fonction `ord()` retourne le code ASCII d'un caractère dans la table.

Exercice 2. Exécuter le script suivant qui affiche à l'écran les caractères de la table ASCII de code allant de 33 à 126 (passage à la ligne tous les 16 caractères) :

```
N = 16
for i in range(33,127):
    print(i, ':', chr(i), end = ' ')
    if (i-32) % N == 0:
        print('')
```

Explication : dans la fonction `print`, l'option `end = ' '` force l'impression en fin de ligne d'un espace `' '` plutôt que d'un caractère de passage à la ligne `end = '\n'`.

Exercice 3. Le chiffrement de César.
C'est une méthode très simple de chiffrement de messages en un texte crypté pour le rendre "illisible" à qui n'en a pas la clef : dans une message changer chaque lettre par la N-ième lettre suivante dans l'ordre alphabétique (après z on reprend en a), où N est un entier entre 1 et 25, c'est la clé de chiffrement.

(1) Ecrire une fonction `Crypt` prenant en paramètre un caractère (chaine d'un seul caractère) et la clé, et retournera le caractère crypté. Par exemple `Crypt('a',3)` renverra le caractère `'d'`, et `Crypt('B',3)` renverra `'E'`.

(2) Ecrire une fontion `CesarCrypt` qui prend deux paramètres une chaine de caractère (le message à crypter) et la clef de chiffrement N, et renvoie le message crypté (chaine de caractère).

(3) Ecrire une fonction `CesarDecrypt` qui prend deux paramètres, un texte crypté et la clef, et retourne le texte décrypté.

Exercice 4. Cryptanalyse du chiffrement de César. On écrit une fonction `CryptAnalyse` qui prend en paramètre un texte, crypté par chiffrement de César, et qu'on souhaite retourner le texte décrypté. Pour cela, pour un texte suffisament long (en français), on peut quasiment toujours supposer que le caractère le plus fréquent est le `'e'`.

(1) Ecrire une fonction qui prend en paramètre une chaine de caractère `txt` et qui renvoie une liste de 26 entiers. A l'indice `i`, la liste contiendra le nombre de fois où le i-ème caractère de l'alphabet minuscule apparait dans la chaine ; par exemple à l'indice 0 le nombre de `'a'` dans `txt`.

(2) Ecrire une fonction `indMax` prenant en paramètre une liste de nombres et renvoie l'indice où se situe le maximum.

(3) Ecrire le code de la fonction `Cryptanalyse` : après avoir récupéré le caractère minuscule le plus fréquent dans la chaine.

(4) Déchiffrer le message crypté, et après avoir affiché son contenu, le sauvegarder dans un fichier `texteDecrypte`.

Exercice 5. Le chiffrement affine.
Pour le chiffrement affine, chaque lettre $A, B, \ldots, Z$ est remplacée par son rang entre 0 et 25.
On choisit deux nombre entiers a et b entre 0 et 25. On note $r(x)$ le reste de la division euclidienne de $y = ax + b$ par 26. La lettre correspondante est la lettre cryptée.

Exemple : avec $a = 11$ et $b = 2$. Pour encrypter D : son rang est 3.
$11 \times 3 + 2 = 35 \equiv 9[26]$. La lettre codée est J.

• **Pour que deux lettres différentes soient cryptées différemment il faut et il suffit que a soit premier avec 26 (*i.e.* $pgcd(a, 26) = 1$).**

(1) Le chiffrement de César correspond à quel cas particulier du chiffrement affine ?

(2) Ecrire une fonction `affCrypt` qui prend en paramètre une lettre et les clés `a` et `b` et retourne le caractère crypté.

(3) Ecrire la fonction `EncryptageAff` qui prend en paramètre le texte à crypter et les clés de cryptage `a` et `b` et retourne le texte crypté.

• **Pour décrypter : appliquer le même principe avec a' et b' tels que :**

$$\begin{cases} aa' \equiv 1[26] \\ b' \text{ **est le reste de la division euclidienne de** } a'(26-b) \text{ **par 26**} \end{cases}$$

(4) Ecrire une fonction `cleInverse` prenant en paramètre `a` et `b` et qui retourne `a'` et `b'`. (On pourra chercher `a'` entre 0 et 25).

(5) Ecrire une fonction `DecryptAff` prenant en paramètre un texte crypté (chaine de caractère) et les clé de cryptage `a` et `b`, et qui retourne le texte décrypté.