TP 2 : Révisions (suite)

1 Chiffre de César

Le chiffre de César est un chiffrement par décalage qui consiste à décaler les lettres de l'alphabet de n lettres avec n fixé et compris entre 0 et 25. Par exemple, pour $n = 3$, on obtient la méthode de chiffrement utilisée par Jules César et illustrée ci-dessous.

non codé	a	b	c	···	w	x	y	z
codé	d	e	f	···	z	a	b	c

Exercice 1

1. Créer une liste `alphabet` contenant toutes les lettres dans l'ordre alphabétique (en minuscule). *Le code ASCII des lettres minuscules est un entier compris entre* 97 *et* 122. `ord('c')` *renvoie l'entier correspondant au code ASCII du caractère* `'c'`, `chr(k)` *renvoie le caractère dont le code ASCII est* `k`.

2. Écrire une fonction `Cesar` prenant pour arguments une chaîne de caractères `m` à chiffrer ainsi qu'un entier `n` correspondant au décalage et retournant la chaîne de caractères chiffrée selon le principe précédent.

   - *On pourra utiliser la méthode* `.index()` *qui retourne la position d'un élément dans une liste ou une chaîne de caractères.*

     ```
     >>> alphabet=['a','b',...,'z']
     >>> alphabet.index('d')
     3
     ```

   - *Les caractères qui ne sont pas des minuscules non accentuées seront ignorés et recopiés.*

3. Tester votre fonction avec `m='allez blaise'` et `n=3`. Vérifier que l'espace et le `z` sont correctement chiffrés.

4. Décoder la chaîne `test1` qui a été chiffrée avec un décalage égal à 5.

   ```
   test1='zs ujz ij ijstrgwjrjsy atzx fyyjsi'
   ```

Exercice 2 Vous trouverez un fichier `test2` à déchiffrer. Vous savez seulement que le chiffre de César a été utilisé mais vous ne connaissez pas le décalage.

1. Créer une fonction `occurrences` qui prend en argument une chaîne de caractères `m` et qui retourne la liste qui contient le nombre d'occurrences de chaque lettre minuscule non accentuée dans `m`. Ainsi, si `L=occurrences(m)`, alors `L[0]` est le nombre de `a` dans `m`, `L[1]` est le nombre de `b` dans `m`,...
   On veillera à ce que la ***complexité*** *pour créer cette liste soit* ***linéaire*** *en la taille de la chaîne de caractères.*

2. a) Créer une fonction `position_max` qui retourne la position de la première apparition du maximum d'une liste d'entiers ou de flottants.
   b) Créer une fonction `decalage_Cesar` qui prend en argument une chaîne de caractère `m` codée à l'aide du chiffre de César et qui retourne le décalage utilisé sachant que la lettre `e` est celle qui apparait le plus fréquemment dans la langue française. Quel est le décalage utilisé pour chiffrer `test2` ?

3. Qui est l'auteur du texte dont vous venez de découvrir la quatrième de couverture ?

Exercice 3 Le chiffre de César est un chiffrement par substitution monoalphabétique. Ceci consiste à substituer dans un message chacune des lettres de l'alphabet par une autre. Voici un exemple :

non codé	a	b	c	···	w	x	y	z
codé	a	z	e	···	c	v	b	n

Combien de tels chiffrements existe-t-il ?

L'analyse des fréquences des lettres permet de casser ces chiffrements.

2 Chiffre de Vigenère

2.1 Chiffrement et déchiffrement

Dans toute cette partie, on suppose que les chaînes manipulées ne comportent que des lettres minuscules non accentuées, ainsi que des espaces et des symboles de ponctuation.
Le chiffre de Vigenère fonctionne sur le principe suivant :

- On dispose d'une clé et d'un texte à coder qui sont tous deux une chaîne de caractères.
- la première lettre du texte est décalée d'un rang correspondant à celui de la première lettre de la clé dans l'alphabet,
- la deuxième lettre du texte est décalée d'un rang correspondant à celui de la deuxième lettre de la clé dans l'alphabet,
- et ainsi de suite...

non codé	j	e		p	r	e	p	a	r	e		l	e	s		c	o	n	c	o	u	r	s
clé	p	y	t	h	o	n	p	y	t	h	o	n	p	y	t	h	o	n	p	y	t	h	o
décalage	15	24	19	7	14	13	15	24	19	7	14	13	15	24	19	7	14	13	15	24	19	7	14
codé	y	c		w	f	r	e	y	k	l		y	t	q		j	c	a	r	m	n	y	g

Exercice 4

1. Écrire une fonction `code_car(c,lettre_cle)` qui prend en argument un caractère et une lettre-clé, et qui renvoie le caractère codé correspondant.

2. Écrire une fonction `Chiffre_Vigenere` prenant comme argument deux chaînes de caractères `cle` et `texte` et effectuant le chiffrement.

3. Vous êtes le destinataire d'un message codé suivant le chiffre de Vigenère et, à ce titre, vous connaissez la clé. Écrire une fonction `Dechiffre_Vigenere` prenant comme argument deux chaînes de caractères `cle` et `texte` et effectuant le déchiffrement. *On pourra au préalable créer une fonction* `decode_car`.

2.2 Comment casser le chiffre de Vigenère

Vous interceptez un message codé à l'aide du chiffre de Vigenère qui ne vous est pas destiné et, à ce titre, vous ne connaissez pas la clé utilisée. La cryptanalyse du chiffre de Vigenère s'effectue en plusieurs étapes.

- Étape 1 : On détermine la longueur de la clé (cf exercice 6).
- Étape 2 : On effectue une analyse des fréquences sur toutes les lettres codées par la même lettre-clé. En effet, si la clé est de longueur N :
  - les lettres d'indice 0, N, $2N$, $3N$,... sont codées par la même lettre-clé,
  - les lettres d'indice 1, $N+1$, $2N+1$, $3N+1$,... sont codées par la même lettre-clé,
  - les lettres d'indice 2, $N+2$, $2N+2$, $3N+2$,... sont codées par la même lettre-clé,
  - ...
- Étape 3 : On recompose la clé et on déchiffre le texte.

Exercice 5 Le texte `test3` a été chiffré par une clé de longueur $N = 8$.

1. Créer la sous-chaîne de caractères des lettres d'indice 0, N, $2N$, $3N$...
2. À l'aide de la fonction `decalage_Cesar`, déterminer la première lettre de la clé utilisée pour chiffrer `test3`.
3. Déterminer la clé en appliquant le principe précédent aux N sous-chaînes de caractères des lettres codées par une même lettre-clé.
4. Déchiffrer le texte `test3`.

Exercice 6 Le test de Kasiski, basé sur l'étude des répétitions dans le message codé, permet d'accéder à la longueur de la clé.

1. Aller lire la page Wikipédia "Cryptanalyse du chiffre de Vigénère" pour comprendre le principe du test de Kasiski.
2. Écrire une fonction `apparitions_motif` qui prend en argument deux chaînes de caractères `mot` et `texte` et qui retourne les rangs des apparitions de la sous-chaîne de caractères `mot` dans `texte`.

3. Écrire une fonction `motifs` qui prend en argument une chaîne de caractères `texte` et un entier `longueur` et qui retourne la liste des motifs de longueur `longueur` qui se répètent dans `texte` ainsi que la liste des listes des distances entre ces répétitions.

4. Déterminer le nombre de distances divisibles par 2, par 3, par 4... entre les répétitions de longueur 3 du texte `test4`.

5. Quelle est la longueur de la clé qui a permis de chiffrer le texte `test4` ? Décoder le texte `test4`.

Exercice 7 Que penser de `test5` et `test6` également codés à l'aide du chiffre de Vigenère ?