L'École Normale Supérieure de l'Enseignement Technique de Mohammedia

Filières Ingénieure

COMPÉTENCES NUMÉRIQUES ET INFORMATIQUE

— TD N°3: —

Fonctions, listes et chaines de caractères

Exercice N° 1(*)

Écrire une fonction de paramètre n, qui définit une liste par compréhension pour ajouter 3 à chaque élément d'une liste d'entiers de 0 à n.

Exercice N° 2(*)

L'objectif est de définir une liste contenant ['ad', 'ae', 'bd', 'be', 'cd', 'ce'] à partir des chaînes "abc" et "de".

a) Créer cette liste par compréhension

b) Simplifier le processus en utilisant une fonction avancée.

Exercice N° 3(**) Médian d'une liste de nombres

a) Écrire la fonction grands(L,x) qui reçoit en paramètres une liste de nombres L, et un élément x de L. La fonction renvoie le nombre d'éléments de L qui sont supérieurs strictement à x.

b) Déterminer la complexité de la fonction grands (L, x), et justifier votre réponse.

c) Écrire la fonction petits(L,x) qui reçoit en paramètres une liste de nombres L, et un élément x de L. La fonction renvoie le nombre d'éléments de L qui sont inférieurs strictement à x.

L est une liste de taille n qui contient des nombres, et m un élément de L. L'élément m est un médian de L, si les deux conditions suivantes sont vérifiées :

- Le nombre d'éléments de L, qui sont supérieurs strictement à m, est inférieur ou égale à n/2
- Le nombre d'éléments de L, qui sont inférieurs strictement à m, est inférieur ou égale à n/2

Exemple : On considère la liste L = [25 , 12 , 6 , 17, 3 , 10 , 20 , 12 , 15 , 38], de taille n=10. L'élément 12 est un médian de L, car :

3 éléments de L sont supérieurs strictement à 12, et $3 \leq n/2$;

5 éléments de L sont inférieurs strictement à 12, et $5 \leq n/2$.

d) Écrire la fonction median(L) qui reçoit en paramètre une liste de nombres L non vide, et qui renvoie un élément médian de la liste L.

Exercice N° 4(**)

Récursivité : Chaînes de caractères

Récursivité et les chaînes :

Une chaîne de caractères **S** est :

- soit vide,
- soit elle peut être vue comme un premier caractère **S[0]** suivi d'une… sous chaîne **S[1 :]**

L'Occurrence d'un caractère d'un mot ou un texte est le nombre d'apparition de ce caractère dans le texte(mot).

a) Écrire une fonction Occur(Text,lettre) et qui retourne le nombre d'occurrences.

b) Proposer une version récursive de la fonction Occur.

Exercice N° 5(*)

Un **palindrome** est un mot dont l'ordre des lettres reste le même si on le lit de gauche à droite ou de droite à gauche. Par exemple : **'laval' , 'radar, 'sos'**... sont des **palindromes**.

Écrire un programme en Python qui demande à l'utilisateur de saisir un mot et de lui renvoyer s'il s'agit d'un palindrome ou non?

Exercice N° 6(**)

Code de César : cryptographie et cryptanalyse Le but de cet exercice est de réaliser un système permettant de crypter des messages, en utilisant le code de César: on part d'un message en clair et on change toutes les lettres en les décalant d'un même nombre de positions (ce nombre est appelé la clé). Pour décoder le message il suffit alors de décaler dans l'autre sens, à condition de connaître la clé.

1. On va commencer par écrire la fonction d'encodage du texte : encrypte(texteEnClair,cle). Toutes les lettres sont cryptées, mais pas les blancs et la ponctuation. Par exemple, ce code affiche les deux lignes notées en commentaire :

```
1 cache=encrypte('Toto est une girafe, avec des petites jambes.',12)
2 print(cache)              # Fafa qef gzq sudmrq, mhqo pqe bqfufqe vmynqe.
3 print(decrypte(cache,12)) # Toto est une girafe, avec des petites jambes.
```

Pour cela, nous allons découper le travail :

a) Écrire une fonction encrypteLettre qui prend une seule lettre en paramètre et renvoie la lettre encodée (attention à la ponctuation et aux

majuscules). Pour décaler les lettres, vous aurez besoin de 2 fonctions de bases de Python : ord et chr.

i. La fonction ord prend une chaine contenant un seul caractère et renvoie son code ascii (ord('a') donne 97, ord('b') 98, etc...).

ii. La fonction chr fait l'inverse : elle renvoie le caractère en fonction du code ascii. Par exemple, le programme suivant permet de créer une liste contenant les lettres de l'alphabet :

```
1 alphabet=[]
2 for i in range(26):
3   alphabet.append(chr(ord('a')+i))
```

b) Écrire une fonction encrypteMot qui prend un mot et renvoie le mot encodé.

c) Enfin écrire encrypte qui renvoie le texte codé.

2. On passe au décodage. Écrire la fonction decrypte(texteCache,cle) qui permet de décoder (et renvoyer) un texte qui a été crypté avec la clé.

3. Passons à la cryptanalyse, c'est-à-dire l'art de déchiffrer les messages sans avoir la clé ! Pour cela, nous allons essayer d'utiliser nos connaissances sur les fréquences des lettres dans un texte. En effet, vous n'êtes pas sans savoir que la lettre la plus commune en français est le 'e'... Donc, pour savoir quelle clé a permis de coder le texte, il suffit de tester toutes les clés possibles et de regarder celle qui donne le texte contenant le plus de 'e'. **Écrire la fonction** devineCle(texteCache) qui renvoie la clé devinée en utilisant ce principe et testez-la sur l'exemple. Attention, cette technique fonctionne d'autant mieux que le texte est long.

4. Cette façon de faire n'est pas très efficace, elle force à décoder 26 fois... Il vaudrait mieux chercher la lettre la plus fréquente dans le texte et chercher la clé permettant de la faire correspondre avec un 'e'. Écrire la fonction devineCle2 qui utilise ce principe. Indice : (une fois que vous connaissez la position pos dans l'alphabet (entre 0 et 25) de la lettre la plus fréquente, il suffit de calculer :

(pos+ord('a')-ord('e'))%26