

CI 2 – ALGORITHMIQUE ET PROGRAMMATION

TP – ALGORITHME DE CÉSAR

Codage de César

Travaux préliminaires

Dans un premier temps, on va charger chacun des mots dans une liste (ou tableau). Pour cela, on définit la fonction suivante :

```
def lire_fichier ( fichier ) :
    """
    Permet de lire un fichier texte en mettant chaque ligne dans une case d'un tableau.
    fichier est une chaine de caracteres contenant le lien vers le fichier .
    """
    # Ouvrir un fichier
    fid=open(fichier , "r")
    tab=[]
    for ligne in fid :
        # Stocker les lignes du fichier dans un tableau en supprimant le retour chariot
        tab.append(ligne.rstrip ('\n\r'))
    # Fermer le fichier
    fid.close()
    return tab

tab= lire_fichier ("liste_mots.txt")
```



Les tableaux permettent de stocker des listes d'éléments. La déclaration d'une se fait ainsi :

```
>>>tab=[0,1,2,3, "a"]
```

La fonction len permet de connaître la taille d'un tableau.

```
>>>len(tab)
```

Les lignes d'un tableau sont numérotées de 0 à len(tab)-1.

Ainsi pour accéder au ième élément d'un tableau on procède ainsi :

```
print (tab[2])
```

Pour parcourir les éléments d'un tableau on utilise la fonction range :

```
for i in range(0, len(tab)) :
    # Suite d' instructions
```



Question 1

Réaliser la fonction `le_mot_le_plus_long` prenant un tableau comme argument et retournant à l'utilisateur le mot plus long.

Question 2

Tester la fonction réalisée. Le résultat correspond-il à vos attentes ? Réaliser la fonction permettant de faire la liste des mots à n lettres.

Question 3

Rechercher le mot ayant le plus grand nombre de lettres.

Question 4

Rechercher un algorithme permettant de faire la liste des mots de moins de n lettres.



Il est possible de réaliser des listes de listes :

```
>>> tab = [["a", 0], ["b", 0], ["c", 0]]
```

On peut alors accéder aux différents items de la liste :

```
>>> print(tab[0][1], tab[1][0])
```

Dans le fichier `ressources.py`, on définit la fonction `recherche_lettre` qui, en fonction d'une lettre, retourne sa position dans l'alphabet. On définit aussi une liste `alphabet` contenant dans chaque élément une lettre ainsi qu'un entier.

Vous pouvez recopier ce qui vous semblera nécessaire dans votre programme.

Question 5

Écrire la fonction `compter_lettre` qui permettra de savoir, à partir du fichier `dictionnaire`, combien de fois sont utilisées chacune des lettres. Cette fonction devra donc retourner une liste comprenant les lettres et leur fréquence d'apparition.

Question 6

Quelle est la lettre qui apparaît le plus dans le dictionnaire ? Donner sa fréquence d'apparition en pourcent.

Codage de César

Le codage de César permet de chiffrer des informations entre un émetteur et un récepteur. Il est nécessaire pour cela de définir une clef numérique connue par les deux interlocuteurs. Cette clef correspond à un décalage des lettres de l'alphabet. Si la clef vaut 1, toutes les lettres de l'alphabet seront décalées d'un cran dans l'ordre alphabétique. Ainsi, le a est remplacé par le b, le b par le c *etc.*.

Pour chiffrer le message, il suffit donc de décaler toutes les lettres. Pour décoder le message lorsqu'on connaît le code il suffit de décaler les lettres dans le sens inverse.

Exemple

Si le message en clair est «allo » et que la clef de chiffrement est 2, le a sera remplacé par le c, le l par le n et le o par le q.
Ainsi le message codé sera «cnnq».

Question 7

Réaliser la fonction permettant de chiffrer un texte suivant la méthode définie ci-dessus.

Question 8

Réaliser une fonction simple permettant de déchiffrer un texte.

Pirate en herbes

Question 9

Trouver une méthode permettant de trouver la clef du code de César.

Question 10

Implémenter cette méthode et donner la source du texte contenu dans le fichier Ressources.py.