Idriss KOUROUMA
NI: 537 025 540

Exercice 1 : Briser le chiffrement par décalage

Question 1 :

La recherche exhaustive de clés consiste à décrypter un message en essayant toutes les clés possibles l'une après l'autre. En l'occurrence, pour le chiffrement par décalage cela revient à essayer tous les décalages (de 1, 2, 3, …, 26 lettres) jusqu'à ce qu'on trouve un message qui fait sens.

Par exemple :

- Testons le décalage de 1 lettre :
  - Lorsque l'on chiffre (lettre + clé) cela revient à passer de A→B, B→C, …, Z→A
  - Lorsque l'on déchiffre (lettre - clé) on passe de A→Z, B→A, …, Z→Y

Pour éviter de faire l'ensemble des tests, j'ai réalisé un programme qui effectue les 26 décalages :

```python
# On déclare l'alphabet qu'on utilise
Alphabet = 'abcdefghijklmnopqrstuvwxyz.,'

# On demande à l'utilisateur de rentrer son texte
texte_chiffre = input('Veuillez rentrez le texte \n')

# On met le texte inscrit par l'utilisateur en minuscule car notre alpahabet est en minuscule
texte_chiffre = texte_chiffre.lower()

for i in range(len(Alphabet)):                          #On testz tous les décalages possibles
    texte_decrypt = ''                                  #On initialise le texte de decryptage comme vide
    print("Chiffrement de Cesar avec un decalage de ", i)   #On indique le décalge où l'on se trouve
    for lettre in texte_chiffre:                        #On parcours le message chiffre
        valeur_lettre = Alphabet.find(lettre)           #On trouve valeur_lettre, qui est par exemple 0 pour A, 1 pour B, ...
        new_val_lettre = (valeur_lettre - i) % len(Alphabet) #new_val_lettre = nouvelle valeur numérique de la lettre, après le décalage de i position(s)
        texte_decrypt = texte_decrypt + Alphabet[new_val_lettre] #On ajoute la nouvelle lettre au texte decrypter
    print (texte_decrypt, "\n")                             #On affiche le texte decrypter
```

On obtient un message qui fait du sens avec un décalage de 22 positions, le message est le suivant :

bonjour,a,tous,et,a,toutes.si,vous,arrivez,a,lire,ceci,c,est,que,vous,avez,reussi,le,premier,challenge.la,prochaine,etape,sera,de,boire,un,bon,petit,caef,relaxer,et,passer,au,prochain,exercice.,desole,pour,les,fautes,d,ortographe.

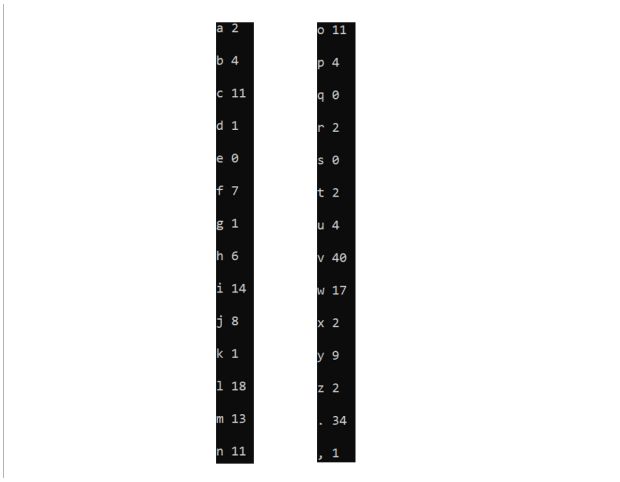
Question 2 :

Pour cette deuxième question, on cherche à effectuer une analyse de fréquences afin de décrypter le message. Le but de l'analyse de fréquence est de trouver la ou les lettres qui sont le plus présentent dans le texte chiffré et de les assimiler aux lettre les plus fréquentes dans l'alphabet d'origine du texte.

Le code suivant permet d'afficher la fréquence des lettres dans une chaîne de caractère donnée. On peut s'en servir afin de trouver la fréquence des différents caractères dans le message chiffré. Il est également possible de la faire à la main, simplement en comptant pour chaque lettre leur nombre d'occurrence.

```python
#question 2
from collections import Counter
Alphabet = 'abcdefghijklmnopqrstuvwxyz'
texte = input("Veuillez saisir le texte \n")
counter = Counter(texte)
for i in Alphabet:
    if counter[i]!=0:
        print(i, counter[i], "\n")
```

On obtient le résultat suivant :



Sachant que les lettres les plus fréquentes dans la langue française sont le $e > a > s > t$

On peut déduire que le $e_x(e) = v$, c'est-à-dire, que la lettre *e* est chiffré par un *v*.

Cela nous donne un décalage de 17. Or d'après la question 1, ce n'est pas le bon décalage. On peut tout de même vérifier :

- Xihdiolvwv (début du texte chiffré) → gtsotzwefe

⇒ Le résultat n'a pas de sens.

On essaye avec une autre association : testons $e_x(e) =$.

⇒ on suppose que le *e* a été codé par le caractère '.'

Ceci donne un décalage de 22, ce qui est le bon décalage.

Si l'on veut décrypter le texte on utilise l'algorithme de la question 1 et on regarde ce qu'il nous renvoie pour clé = 22. On s'aperçoit qu'on retrouve bien le même message que trouvé en question 1.