	**Architecture matérielle**	**NSI T^{ale}**
	Chiffrement par substitution	TP

*

1. Chiffrement par substitution

Le **chiffrement par substitution simple** consiste à **remplacer dans un message chacune des lettres de l'alphabet par une autre.**

Par exemple, si la lettre **m** est remplacée par **t** et la lettre **i** est remplacée par la lettre **o** alors le mot **mimi** est remplacé par **toto**.

La substitution est définie par une **permutation** des lettres de l'alphabet. Nous nous intéressons ici au décodage des messages chiffrés par substitution. Nous nous limiterons aux **messages écrits en français sans accent, sans majuscule, sans espace et sans ponctuation**. L'alphabet utilisé comporte donc 26 lettres minuscules.

2. Chiffrement d'un message par substitution

Le programme suivant permet de faire le chiffrement d'un message par substitution en utilisant un alphabet de substitution aléatoire.

```
from random import shuffle
import string

texte = "enessayantcontinuellementonfinitparreussirdoncpluscarate-
plusonadechancesquecamarche"

texteChiffre = ""

dicoSubstitution = {}

alphabet = list(string.ascii_lowercase.strip())
#OU [chr(i) for i in range(97,123)]
#print(alphabet)

#création de l'alphabet à l'aide de la fonction shuffle effectuant
une permutation aléatoire
alphabetSubstitution=list(string.ascii_lowercase.strip())
shuffle(alphabetSubstitution)
#Affichage de l'alphabet de substitution
#print(alphabetSubstitution)
```

```
#création du dictionnaire de substitution
for i in range(26):
    dicoSubstitution[alphabet[i]]=alphabetSubstitution[i]
#Affichage du dictionnaire de substitution
print(dicoSubstitution)

#Création du message chiffré
for lettre in texte:
    texteChiffre += dicoSubstitution.get(lettre)
print(texteChiffre)
```

3. Nombre de chiffrements par substitution

Chaque permutation des 26 lettres fournit une possibilité de chiffrement. Pour décoder un message chiffré par substitution, on pourrait lister toutes les permutations possibles.

Le nombre de permutations est égal à 26×25×24×...×1 = 26!

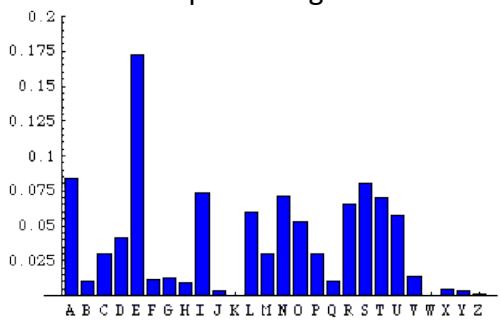
Q1. a. Compléter la fonction suivante qui permet de calculer la factorielle d'un entier naturel *n* (elle prend en paramètre un entier naturel *n* et renvoie *n*!)

```
def factorielle(n):
    """
        Calcul la factorielle d'un entier naturel n

        :param n: le nombre dont on veut caluler la factorielle
        :type n: int
        :return: factorielle de n
        :rtype: int

        :Example:

        >>> factorielle(0)
        1
        >>> factorielle(10)
        3628800
    """
    fact = 1

    for i in range(..........):
        fact =.....
    return fact

print(factorielle.__doc__)
```

Q1. b. Tester votre fonction `factorielle` pour donner le nombre de permutations possibles pour un alphabet de 26 lettres. Conclure sur la faisabilité de la méthode de déchiffrement basée sur l'examen de tous les chiffrements possibles.

4. Utilisation de la fréquence d'apparition des lettres

Pour **déchiffrer un message codé par substitution**, il est plus efficace d'utiliser l'**analyse fréquentielle**. Celle-ci utilise le fait que, dans une langue donnée, la fréquence d'apparition d'une lettre donnée varie peu d'un texte à l'autre.
La **répartition des lettres en français** est donnée par le diagramme suivant :



Source : https://www.apprendre-en-ligne.net/crypto/stat/francais.html

5. Détermination de la fréquence d'une lettre dans un texte

La fonction **`frequenceLettre`** prend en paramètres une lettre et un texte (chaînes de caractères) en paramètres et renvoie la fréquence d'apparition de la lettre dans le texte.

Q2. a. Compléter la fonction `frequenceLettre` ainsi que sa docstring

```
def frequenceLettre(lettre,texte):
    """


    :param :
    :type :
    :return:
    :rtype:

    :Example:

    >>>

    >>>

    """
    compteur = 0
    for l in texte:
        if ..............
            compteur.................
    return ...................

print(frequenceLettre.__doc__)
```

Q2. b. Tester votre fonction pour le texte suivant extrait de Cyrano de Bergerac (sans majuscule, accent, espace ou ponctuation) et la lettre a.

Faire afficher le résultat pour le texte tronqué de 206 caractères et pour le texte entier
Si votre fonction ne renvoie pas environ 0,087 pour le texte tronqué de 206 caractères, il vous faudra la re-travailler...

```
texte = "ahnoncestunpeucourtjeunehommeonpouvaitdireohdieubiendescho-
sesensommeenvariantletonparexemplетenezagressifmoimonsieursijavai-
suntelnezilfaudraitsurlechampquejemelamputasseamicalmaisildoittrem-
perdansvotretassepourboirefaitesvousfabriquerunhanapdescriptifces-
tunroccestunpiccestuncapquedisjecestuncapcestunepeninsulecurieuxde-
quoisertcetteoblonguecapsuledecritoiremonsieuroudeboiteaciseauxgra-
cieuxaimezvousacepointlesoiseauxquepaternellementvousvouspreoccupa-
tesdetendreceperchoiraleurspetitespattestruculentcamonsieurlorsque-
vouspetunezlavapeurdutabacvoussortelledunezsansquunvoisinnecrieau-
feudechemineeprevenantgardezvousvotreteteentraineeparcepoidsdetombe-
renavantsurlesoltendrefaitesluifaireunpetitparasoldepeurquesacouleu-
rausoleilnesefanepedantlanimalseulmonsieurquaristophaneappellehippo-
campelephantocamelosdutavoirsouslefronttantdechairsurtantdoscava-
lierquoiiamicecrocestalamodepourpendresonchapeaucestvraimenttrescom-
modeemphatiqueaucunventnepeutnezmagistraltenrhumertoutentierexcepte-
lemistraldramatiquecestlamerrougequandilsaigneadmiratifpourunparfu-
meurquelleenseignelyriqueestceuneconqueetesvousuntritonnaifcemonu-
mentquandlevisitetonrespectueuxsouffrezmonsieurquonvoussaluecestla-
cequisappelleavoirpignonsurruecampagnardheardecestyunneznanaincest-
queuqunavetgeantoubenqueuqumelonnainmilitairepointezcontrecavalerie-
pratiquevoulezvouslemettreenloterieassurementmonsieurceseralegroslo-
tenfinparodiantpyrameenunsanglotlevoiladonccenezquidestraitsdeson-
maitreadetruitlharmonieilenrougitletraitrevoilacequapeupresmoncher-
vousmauriezditsivousaviezunpeudelettresetdespritmaisdespritoleplus-
lamentabledesetresvousneneutesjamaisunatomeetdelettresvousnavezque-
lestroisquiformentlemotsoteussiezvouseudailleursiinventionquilfaut-
pourpouvoirladevantcesnoblesgaleriesmeservirtoutescesfollesplaisan-
teriesquevousneneussiezpasarticulelequartdelamoitieducommencementdu-
necarjemelessersmoimemeavecassezdevervemaisjenepermetspasquunautre-
melesserve"

print("Nombre de caractères dans le texte",len(texte))
print("Fréquence des a le texte",frequenceLettre("a",texte))
```

6. Fluctuation de la fréquence d'apparition d'une lettre en fonction de la longueur du texte

Importation de la bibliothèque graphique.

```
import matplotlib.pyplot as plt
```

Pour des textes longs, la fréquence d'apparition observée est proche d'une valeur qui peut s'interpréter comme **fréquence d'apparition de la lettre dans la langue française**.
C'est pourquoi, nous allons étudier le texte précédant comprenant 1917 caractères.

Le programme suivant permet de représenter la fréquence d'apparition de la lettre `lettre_etudiee` dans un extrait de ce texte en fonction de la longueur de l'extrait.

```
if plt.get_fignums():
    plt.clf() #pour effacer la figure

#nous travaillons avec le texte précédent
N = len(texte)
frequences = []

lettre_etudiee = "a"
#pour des textes allant de 0 à 1917 caractère
for i in range(1, N):
    #permet d'extraire les i premiers caractères du texte
    frequences.append(frequenceLettre(lettre_etudiee, texte[:i]))

plt.title("Fréquence d'apparition de la lettre "+lettre_etudiee+" en
fonction de la longueur du texte",color="#1e7fcb",fontsize=11)
plt.plot(frequences)
plt.show()
```

Q3. Modifier le programme précédent pour afficher sur le même graphique, les fréquences d'apparition des lettres c, h, i, f, r et e

7. Liste des fréquences des lettres de l'alphabet dans un texte donné

Q4. Compléter la fonction `dicoFrequences` (et son docstring) qui renvoie les fréquences des 26 lettres de l'alphabet dans un texte donné.

Elle prend en paramètre une **chaîne de caractères** et renvoie un **dictionnaire** contenant la fréquence d'apparition des lettres de l'alphabet (les **clés sont les lettres de l'alphabet**) de la chaîne de caractères.

```
def dicoFrequences(texte):
    """

    :param :
    :type :
    :return:
    :rtype:

    :Example:

    >>>

    >>>

    """
    alphabet =.............
    dico =.............
    for l in alphabet:
        ..................................
    return dico

print(dicoFrequences(texte))
```

8. Diagramme en bâtons des fréquences d'apparition des lettres de l'alphabet

Q5. a. Écrire le code qui permet de créer une liste des fréquences à partir du dictionnaire précédemment obtenu.

```
#on transforme le dictionnaire en liste : on ne conserve que les frequences
dicoFreq = dicoFrequences(texte)
listeFreq = ............................
print(dicoFreq, listeFreq)
```

Q5. b. Compléter le programme suivant permet de représenter les fréquences des lettres apparaissant dans le texte sous la forme d'un diagramme en bâtons.

```
if plt.get_fignums():
    plt.clf() #pour effacer la figure

epaisseur = 0.5
alphabet = ......................

plt.bar(.........................................)
plt.xlabel('Lettres')
plt.ylabel('Fréquences')
plt.title("Fréquences d'apparition des lettres dans le texte")
plt.show()
```

9. Fréquences d'apparition des lettres dans la langue française

Le programme suivant représente la fréquence d'apparition des lettres dans les textes en langue française. Il ne s'agit ici que d'estimations. La nature du texte peut en effet faire varier légèrement ces fréquences. La liste donnant ces fréquences pourra être donnée ou remplacée par une autre (Wikipedia).

L'objectif est ici de comparer cette représentation à la précédente.

```
if plt.get_fignums():
    plt.clf() #pour effacer la figure

alphabet = [chr(i) for i in range(97,123)]

frequencesFrance=[0.077,0.013,0.033,0.042,0.174,0.012,0.013,0.011,0.
071,0.003,0.005,0.061,0.029,0.064,0.051,0.029,0.007,0.061,0.092,0.07
1,0.057,0.013,0.001,0.004,0.005,0.001]
#listeFreq

epaisseur = 0.3

barresFR = range(len(alphabet))
barresTXT = [x + epaisseur for x in barresFR]

plt.bar(barresFR, frequencesFrance, epaisseur, color='r',label='Fréq
dans la langue française')
plt.bar(barresTXT,  ..............................,label='Fréq  dans
le texte')
plt.legend()
plt.xticks([r + epaisseur / 2 for r in range(len(barresFR))], alpha-
bet)

plt.xlabel('Lettres')
plt.ylabel('Fréquences')
plt.title("Comparaison des fréquences d'apparition des lettres")
plt.show()
```

10. Message chiffré et déchiffrement

Q6. Compléter le programme suivant qui permet de déchiffrer un message chiffré avec la correspondance représentée par le dictionnaire `dictChiffrage` donné.

```
#dictionnaire de correspondance pour le chiffrement du message
dictChiffrage = {'a': 'n', 'b': 'g', 'c': 't', 'd': 'z', 'e': 'x',
'f': 'l', 'g': 'm', 'h': 'd', 'i': 'w', 'j': 's', 'k': 'p', 'l':
'h', 'm': 'b', 'n': 'u', 'o': 'e', 'p': 'k', 'q': 'v', 'r': 'i',
's': 'j', 't': 'q', 'u': 'a', 'v': 'f', 'w': 'r', 'x': 'y', 'y':
'c', 'z': 'o'}

texteChiffre = "ndueutxjqaukxateaiqsxauxdebbxeukeafnwqz-
wixedzwxagwxuzxjtdejxjxujebbxxufniwnuqhxqeuknixyxbkhxqxuxonmixjjwl-
bewbeujwxaijwsnfnwjauqxhuxowhlnazinwqjaihxtdnbkvaxsxbxhnbkaqnj-
jxnbwtnhbnwjwhzewqqixbkxiznujfeqixqnjjxkeaigewixlnwqxjfeajlngiw-
vaxiaudnunkzxjtiwkqwltxjqauiettxjqaukwttxjqautnkvaxzwjsxtxjqautnktx-
jqauxkxuwujahxtaiwxayzxvaewjxiqtxqqxegheumaxtnkjahxzxtiwqewixbeu-
jwxaieazxgewqxntwjxnaymintwxaynwbxofeajntxkewuqhxjewjxnayvaxkn-
qxiuxhhxbxuqfeajfeajkixettaknqxjzxqxuzixtxkxitdewinhxaijkxqwqxjknq-
qxjqiatahxuqtnbeujwxaiheijvaxfeajkxqauxohnfnkxaizaqngntfeajjeiqxhhx-
zauxojnujvaaufewjwuuxtiwxnalxazxtdxbwuxxkixfxunuqmnizxofeajfeqixqx-
qxxuqinwuxxknitxkewzjzxqebgxixunfnuqjaihxjehqxuzixlnwqxjhawln-
wixaukxqwqkninjehzxkxaivaxjnteahxainajehxwhuxjxlnuxkxznuqhnuwbnh-
jxahbeujwxaivaniwjqekdnuxnkkxhhxdwkketnbkxhxkdnuqetnbxhejzaqnfewi-
jeajhxlieuqqnuqzxtdnwijaiqnuqzejtnfnhwxivaewwnbwtxtietxjqnhnbezx-
keaikxuzixjeutdnkxnatxjqfinwbxuqqixjtebbezxxbkdnqwvaxnataufxuquxkxa-
quxobnmwjqinhqxuidabxiqeaqxuqwxixytxkqxhxbwjqinhzinbnqwvaxtxj-
qhnbxiieamxvanuzwhjnwmuxnzbwinqwlkeaiauknilabxaivaxhhxxujxwmuxhciw-
vaxxjqtxauxteuvaxxqxjfeajauqiwqeuunwltxbeuabxuqvanuzhxfwjwqx-
qeuixjkxtqaxayjeallixobeujwxaivaeufeajjnhaxtxjqhntxvawjnkkxhhxnfe-
wikwmueujaiiaxtnbknmunizdxnizxtxjqcauuxoununwutxjqvaxavaunfxqmxnu-
qeagxuvaxavabxheuunwubwhwqnwixkewuqxoteuqixtnfnhxiwxkinqwvaxfeahxo-
feajhxbxqqixxuheqxiwxnjjaixbxuqbeujwxaitxjxinhxmiejheqxulwukniezwnu-
qkcinbxxuaujnumheqhxfewhnzeuttxuxovawzxjqinwqjzxjeubnwqixnzxqiawqhd-
nibeuwxwhxuieamwqhxqinwqixfewhntxvankxakixjbeutdxifeajbnaiwxozwqjw-
feajnfwxoaukxazxhxqqixjxqzxjkiwqbnwjzxjkiwqehxkhajhnbxuqnghxzxjxqix-
jfeajuxuxaqxjsnbnwjaunqebxxqzxhxqqixjfeajunfxovaxhxjqiewjvawleibxu-
qhxbeqjeqxajjwxofeajxaznwhhxaijwwufxuqweuvawhlnaqkeaikeafewihnzxfnu-
qtxjueghxjmnhxiwxjbxjxifwiqeaqxjtxjlehhxjkhnwjnuqxiwxjvaxfeajuxuxaj-
jwxoknjniqwtahxhxvaniqzxhnbewqwxzatebbxutxbxuqzauxtnisxbxhxjjxij-
bewbxbxnfxtnjjxozxfxifxbnwjsxuxkxibxqjknjvaaunaqixbxhxjjxifx"

texteDechiffre = ""

#création du dictionnaire de déchiffrage : on inverse clés et va-
leurs
dictDechiffre = ..........................
print(dictDechiffre)
```

```
#déchiffrage du texte chiffre
.............................

print(texteDechiffre)
```

Q7. Le message ci-dessous a été chiffré à l'aide d'une substitution simple à partir d'un message écrit en français : déchiffrer le message en étudiant les fréquences d'apparition des lettres du message (à la main)

rertepadmneppobewogexaemefopwtvvdpexeaweaxdtxeofxepwerzwobe

Les fréquences aident au déchiffrement mais ne suffisent pas dans le cas d'un message court. En effet, la fluctuation liée à la taille de l'échantillon est importante.

Q8. Vérifier les fréquences des lettres dans le message ci-dessus en affichant le graphique.