Partie 1

Exercice 1 : Substitution monoalphabétique : Code de César

Le chiffre de César fonctionne par décalage des lettres de l'alphabet.

Cet algorithme de chiffrement utilise une fonction périodique pour transformer les rangs de chaque lettre. On utilise l'alphabet suivant :

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

- **Question 1** : Compléter l'algorithme du chiffrement de César. On suppose qu'il existe une clé `c` correspondant au décalage utilisé. Le message clair est `m` et le message chiffré est `m_chiffre`. On dispose de 2 fonctions, `rang` qui retourne le rang d'un caractere `c` dans l'alphabet, et `lettre` qui retourne le caractère correspondant au rang `i`.

```
Pour chaque lettre l du message m:
  l_chiffre = ...
  m_chiffre = ...
```

- **Question 2** : Quelle clé a été utilisée pour chiffer ce texte (avec l'algorithme de César) ?

$$jyfwavnyhwoplhwwspxbll$$

- **Question 3** : Décrypter ce message.
- **Question 4** : Proposer un programme en python qui chiffre un message clair `m` en un message codé selon la clé numérique `cle`. Programmer les fonctions `rang`, `lettre`, ainsi que la fonction de chiffrement `chiffrer`.

Aide :

- La fonction `ord('x')` retourne un entier correspondant à la position d'un caractère dans la table ascii. Par exemple, `ord('A')` retourne 65, `ord('B')` retourne 66,...
- La fonction `chr(N)` retourne le caractère de rang N : `chr(65)` retourne 'A'.

Partie 2

Substitution polyalphabétique : Chiffrement de Playfair

A partir des explications données dans la video :

- **Question 1** : Construisez la matrice pour l'algorithme de Playfair avec la clé *estienne*.
- **Question 2** : Chiffrer le message : *ACTIONREACTION*
- **Question 3** : S'agit-il d'une méthode utilisant la substitution monoalphabetique, ou polyalphabetique ?
- **Question 4** : Est-ce qu'avec cette méthode, le decryptage peut être facilité par l'analyse fréquentielle ?

Partie 3

Frequences des lettres dans un texte

La fonction suivante retourne une liste de 26 valeurs de type *float*, donnant dans l'ordre de l'alphabet, la frequence pour chaque lettre dans un texte :

```
def freq(m):
    frequences = [0]*26
    n = len(m)
    for c in m:
    ...
```

- **Question 1** : Compléter le script de la fonction `freq`.

Soient les deux textes chiffrés suivants :

a. DGFHTMOMEIAMTLMFGOKFGOKEGDDTRWEIAKZGFGFSTEKGOMLASTTIFG FOSTLM FTFGOK MGWMFG OKRTSA JWTWTA WDTFMG FDAOLT WMOSSA FGOKET WKRWFD TEIAFM ROAZSG MOFKOT FFTXAW MLARGW ETWKJW AFROSD OAWSTA WDAMOF HGWKDT STEITK SADAOF

b. YOHGMV MFCBRB GWFNIZ ZPSURW FUOIPU WYITFA NIETGG DOCKAC PQE- BEW PMXEMK VGRAIL KWWXZO CILGPM QOVCGE GMYEBQ RYACJM WX- ULFR VQMDCY LZFYZM YTPCRF DCRJNS FIHIQG RZEPRC VWMDIL KGYDQO RVFMXM CRCNIM UGRBKR BOOIUG PQCBVZ NEYACE

- **Question 2** : L'un des 2 a été chiffré avec un algorithme de substitution mon-alphabetique, et l'autre, poly-alphabétique. Lequel est mono-alphabétique ?

Partie 4

Chiffrement symétrique par la fonction XOR

- **Question 1** : La fonction XOR est la fonction du OU EXCLUSIF.

Donner la table de vérité de la fonction XOR

- **Question 2** : A partir des lignes suivantes, vérifier que l'opérateur ^ en python réalise la fonction XOR sur 2 nombres écrits en valeur décimale :

```
> 3 ^ 4
7
> 4 ^ 5
1
```

- **Question 3** : Programmer la fonction `xor` en python qui retourne le resultat de XOR appliqué aux 2 paramètres a et b.

Exemple :

```
>>> xor(53,23)
34
>>> xor(34,23)
53
```

- **Question 4** : Peut-on utiliser la fonction XOR pour chiffrer PUIS dechiffrer un message, avec la MEME clé de chiffrement ? Expliquer à partir de l'exemple ci-dessus.
- **Question 5** : On s'interresse maintenant à la programmation de la fonction XOR pour chiffrer un message à partir d'un masque :

$$message = "CETTEPHRASEESTVRAIMENTTRESTRESLONGUEMAISCESTFAITEXPRES"$$

Le masque (correspond à la clé) aura une longueur inférieur au message. On utilise le masque selon la méthode de chiffrement polyalphabetique vue en cours.

Le chiffrement demande de faire correspondre les caractères avec des nombres entiers. On utilisera pour cela un alphabet de 32 caractères (2^5). Chaque lettre de l'alphabet correspondra alors à un nombre compris entre 0 et 31.

- **Question 6** : Lorsque l'on place 2 entiers inférieurs à 32 dans la fonction XOR, celle-ci retourne un entier également inférieur à 32. Pourquoi ? *(Representer ces nombres en binaire, sur un octet, et verifier cette affirmation).*
- **Question 7** : Compléter la fonction `chiffre(message, masque)` qui chiffre message en le XORant avec masque.

Cette fonction doit pouvoir aussi servir à déchiffrer le message chiffré.

```
alphabet=[chr(i+ord('A')) for i in range(26)] + ['*','/','@','&','!','à
    ']
def chiffre(message, masque):
    assert len(message)>=len(masque)
    message_chiffre = ""
    for i in range(len(message)):
        indice_lettre = alphabet.index(message[i])
        indice_masque = ...
        val = xor(...,...)
        message_chiffre+=alphabet[val]
    return ...
```

- **Question 8** : Utiliser la fonction `chiffre` pour effectuer le chiffrement puis le dechiffrement du *message*.
- **Question 9** : Pensez vous que cette méthode de chiffrement est assez resistante face à une attaque de cryptoanalyse ? Si oui, pourquoi ?