TP0 Chiffrement et déchiffrement de messages

Nous avons reçu ce message codé suivant et sommes chargés de **mettre en place un programme de déchiffrement**.

```
ODQHGHVYVHZRB IZAF YUZHF FDABBH Y EDGMHOOFDF GD NDBBYLD EYRB GD VI GZNIQDV JAH
 GZRBVHVAYHV ARD FDUHBHZR ED RZNPFDABDB RZVHZRB FDRGZRVFDDB DR IFDNHDFD YRRDD
OZRGVHZRB QHBVDB NYRHIAQYVHZR ED OHGMHDFB LFYIMHJADB  RZAB YPZFEDFZRB GDVVD YR
RDD QDB RZVHZRB ED PYBDB ED EZRRDDB EHGVHZRRYHFDB IFZLFYNNYVHZR ESRYNHJAD VMDZ
FHD ED KDAW DV HRVDQQHLDRGD YFVHOHGHDQQD RZVYNNDRV QDB PYBDB ED Q YIIFDRVHBBYL
D YAVZNYVHJAD DR PFDO AR UYBVD IFZLFYNND
```

Nous savons que l'émetteur du message a utilisé un chiffrement par substitution : chaque lettre de l'alphabet a été remplacée par une autre. Nous devons donc retrouver la clé de chiffrement afin de traduire le message.

Remarque : Au cours de l'histoire, la méthode de chiffrement par substitution a été très utilisée : du chiffrement de César pour l'envoi de missives codées, à la machine Enigma pendant la seconde guerre mondiale.

Attention au vocabulaire : Dans le but de rendre un fichier lisible, on parle de déchiffrement et non de décryptage considéré comme incorrect (idem pour chiffrement vs cryptage).

Pour ce TP, il est évident que chacune des fonctions doit être testée au fur et à mesure, pour valider son bon fonctionnement.

Etape 1 : Chiffrement par substitution

Dans cette étape, nous allons développer une machine à chiffrer les messages. Pour simplifier le problème, les textes que nous traiterons ne comporteront que des caractères et des espaces (ni ponctuation, ni accent). L'objectif est de créer un encodeur qui transforme une chaine de caractères en une liste d'entiers.
Un encodage consiste à remplacer l'ensemble des caractères d'un texte par d'autres.
Notre première mission est de coder les caractères par leur position dans l'alphabet. De manière arbitraire, on choisit que l'espace est codé par l'entier 0.

Ainsi :

espace	A	B	C	…	Z
0	1	2	3	…	26

1.1. Ecrire une fonction **caractereVersEntier** qui prend pour paramètre un caractère et retourne un entier qui correspond à la position du caractère dans l'alphabet. Comme choisi précédemment, si le caractère est un espace, l'entier retourné vaudra 0.

Exemples :

```
print( caractereVersEntier ('C') ) # affiche 3
print( caractereVersEntier (' ') ) # affiche 0
```

Vous pouvez utiliser la fonction ord de Python.
(cf documentation en ligne pour utilisation : https://docs.python.org/fr/3/library)

1.2. Ecrire une fonction **entierVersCaractere** qui prend pour paramètre un entier position et retourne le caractère qui correspond à la position du caractère dans l'alphabet.

Exemples :

```
print( entierVersCaractere (3) ) # affiche C
print( entierVersCaractere (0) ) # affiche un espace
```

Vous pouvez utiliser la fonction chr de Pyhton.

1.3. Ecrire une fonction **encode** qui prend pour paramètre une chaine de caractères texte et retourne une liste d'entiers correspondant aux positions successives de chaque caractère de la chaine texte en entrée.

Exemples :

```
print( encode('BONJOUR') )  # affiche [2, 15, 14, 10, 15, 21, 18]
print( encode('BON JOUR') ) # affiche [2, 15, 14, 0, 10, 15, 21, 18]
```

1.4. Ecrire une fonction la fonction inverse **decode**.

Exemples :

```
print( decode([2, 15, 14, 10, 15, 21, 18]) )     #affiche BONJOUR
print( decode([2, 15, 14, 0, 10, 15, 21, 18]) )  #affiche BON JOUR
```

1.5. Une clé de chiffrement peut être représentée par une fonction f qui à chaque lettre de k associe une autre lettre f(k). f doit être bijective : chaque lettre est codée par une unique autre lettre et f(0) = 0 pour gérer les espaces. f sera représentée par une liste de 27 entiers dans laquelle le k^{ème} élément de la liste corresponde à la k^{ème} lettre.

Ecrire une fonction **genereCode** qui renvoie une liste de 27 entiers distribués aléatoirement. Vous pouvez utiliser la fonction Python shuffle.

Exemple :

```
print ( genereCode() ) # exemple de generation
#[0, 5, 13, 3, 2, 25, 19, 9, 20, 16, 14, 15, 8, 4, 6, 22, 18, 1, 23, 26, 10, 2
1, 11, 24, 17, 12, 7]
```

1.6. Ecrire une fonction **chiffrer** qui :

- prend en paramètres une chaine texte et la clé de chiffrement cle sous forme de liste d'entiers,
- et renvoie la chaine de caractères chiffrée.

Exemple :

```
cle = [0, 5, 13, 3, 2, 25, 19, 9, 20, 16, 14, 15, 8, 4, 6, 22, 18, 1, 23, 26,
10, 21, 11, 24, 17, 12, 7]
print ( chiffrer('BONJOUR', cle) ) # affiche MVFNVUW
```

1.7. Ecrire une fonction **dechiffrer** qui :

- prend en paramètres une chaine texte et la clé de chiffrement cle sous forme de liste d'entiers,
- et renvoie la chaine de caractères déchiffrée.

Exemple :

```
cle = [0, 5, 13, 3, 2, 25, 19, 9, 20, 16, 14, 15, 8, 4, 6, 22, 18, 1, 23, 26,
10, 21, 11, 24, 17, 12, 7]
print ( dechiffrer('MVFNVUW', cle) ) # affiche BONJOUR
```