Informatique en CPGE (2017-2018)
Corrigé TD 7 : codage

1 Chiffrement de César

C'est un chiffrement par décalage.

Le texte chiffré s'obtient en remplaçant chaque lettre du texte original par une lettre à distance fixe, toujours du même côté, dans l'ordre de l'alphabet.

Exercice 1

1. Définir une chaîne de caractères contenant toutes les lettres dans l'ordre alphabétique (caractères en minuscule).

```
alphabet='abcdefghijklmnopqrstuvwxyz'
```

2. Ecrire une fonction **decalage**, d'argument un entier **n**, renvoyant une chaîne de caractères contenant toutes les lettres dans l'ordre alphabétique, décalées de **n**, comme indiqué ci-dessus.

```
def decalage(n):
    n=n%26
    return alphabet[n:26]+alphabet[0:n]
```

3. Ecrire une fonction **indices**, d'arguments un caractère **x** et une chaîne de caractères **phrase**, renvoyant une liste contenant les indices de **x** dans **phrase** si **x** est une lettre de phrase et une liste vide sinon.

```
def indices(x,phrase):
    return [i for i in range(len(phrase)) if phrase[i]==x]
```

4. Ecrire une fonction **codage** d'arguments un entier **n** et une chaîne de caractères **phrase**, renvoyant **phrase** codé avec un décalage de **n** lettres.

```
def codage(n,phrase):
    phrase_code=''
    alphabet_decale=decalage(n)
    for lettre in phrase:
        position=indices(lettre,alphabet)
        if len(position)!=0:
            phrase_code+=alphabet_decale[position[0]]
        else:
            phrase_code+=lettre
    return phrase_code
```

5. Comment peut-on décoder un mot codé si on connaît le décalage ? Et si on ne le connaît pas ?

   Si on connaît n :

```
phrase_code=codage(n,phrase)

print(codage((26-n)%26,phrase_code))
```

   Si on ne connaît pas la valeur de n, pour une phrase courte, on teste toutes les possibilités :

```
    for i in range(26):
        print(codage((26-i)%26,phrase_code))
```

Pour une phrase longue on peut faire une analyse de fréquences avec la fonction indices et on suppose que la lettre le plus souvent rencontrée est un e, ou un s ou un u (en français)"

```
    def frequence(phrase):
        maxi=0
        lettre_max=''
        for lettre in alphabet:
            if len(indices(lettre,phrase))>maxi:
                maxi=len(indices(lettre,phrase))
                lettre_max=lettre
        return lettre_max,maxi
    print("lettre la plus frequente",frequence(phrase))
```

On calcule alors le décalage entre 'e' et maxi, ou entre 's' et maxi,...

Utilisation de la table ASCII comme alphabet
Décoder la suite de nombres : 98 114 97 118 111

```
print(chr(98)+chr(114)+chr(97)+chr(118)+chr(111))
# affiche bravo
```

2 Chiffrement affine

Exercice 3

1. (a) Coder la lettre U.

```
alphabet="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
def code(lettre):
    m=alphabet.index(lettre)
    p=(9*m+5)%26
    return alphabet[p]
print(code("U")) # affiche D
```

(b) Ecrire une fonction qui prend en argument un nombre m entier naturel, et renvoie le nombre p, calculé à l'aide du procédé de codage précédent.

```
def f(m):
    return (9*m+5)%26
```

2. (a) Trouver un nombre entier x tel que $9x \equiv 1 \quad [26]$.

```
x=0
while x<26:
    if (9*x)%26==1:
        print(x)
        break
    else :
        x+=1
```

(b) Démontrer alors l'équivalence :

$$9m + 5 \equiv p \quad [26] \iff m \equiv 3p - 15 \quad [26].$$

(c) Décoder alors la lettre B.

```
def decode(lettre):
    p=alphabet.index(lettre)
    m=(3*p+11)%26
    return alphabet[m]
print(decode("B")) # affiche O
```

3. Ecrire un programme permettant de coder un message entré par l'utilisateur, puis de le décoder. On pourra utiliser le code ASCII ou pas.

3 Chiffrement de Vigenère

Exercice 4

1. Définir le texte à coder et la phrase clé, par exemple :

```
texte="Hello my friend, you are going to decode my message"

phrase_cle="Attention, le texte est en anglais !"
```

2. Ecrire une fonction qui prend en argument deux chaînes de caractères, le texte à coder et la phrase clé, et renvoie le texte codé. Pour chaque lettre du texte, la clé sera donnée par la fonction **ord** appliquée à chaque lettre de la phrase clé.

```
def codage(texte,phrase_cle):
    texte_code=''
    longueur=len(phrase_cle)
    for i in range(len(texte)):
        j=i%longueur
        cle=ord(phrase_cle[j])
        n=ord(texte[i])
        if n>=97 and n<=122:
            n=((n-97)+cle)%26+97
        if n>=65 and n<=90:
            n=((n-65)+cle)%26+65
        print(i,j,phrase_cle[j],cle,chr(n))
        texte_code+=chr(n)
    return texte_code
```