Chiffrement

Nous utiliserons dans ce TP le langage de programmation *Python* au moyen de l'environnement de développement *IDLE*.

Exercice 1. On considère le programme suivant :

```
from tkinter import*
# récupération du nom du fichier texte au moyen d'une boîte de dialogue
fenetre=Tk()
nomfichier=filedialog.askopenfilename(filetypes=[("fichier texte",".txt")])
fenetre.destroy()
# lecture du fichier
fichier=open(nomfichier,'r')
chaine=fichier.read()
fichier.close()
# conversion en majuscule
CHAINE=chaine.upper()
# écriture dans le fichier
fichier=open(nomfichier,'w')
fichier.write(CHAINE)
fichier.close()
```

Créer un fichier texte puis tester sur celui-ci le programme précédent.

Exercice 2. L'algorithme de chiffrement *ROT13* consiste à décaler chaque lettre d'un message de treize crans dans l'alphabet :

$$\begin{array}{ccc} a & \mapsto & n \\ b & \mapsto & o \\ & \vdots & \\ y & \mapsto & l \\ z & \mapsto & m \end{array}$$

Créer un programme Python permettant de réaliser le chiffrement ROT13 d'un fichier texte, puis créer un programme Python permettant de réaliser le déchiffrement.

Exercice 3. Les algorithmes de chiffrement à décalage fixe comme le ROT13 sont relativement faciles à casser, en effet, pour un alphabet de 26 caractères, il n'existe que 26 décalages possibles (dont un trivial) et il suffit de tous les tester pour déchiffrer le message considéré.

L'algorithme de *Vigenère* consiste à introduire une clef qui permet de rendre le décalage variable selon la position du caractère dans le message.

L'exemple ci-dessous effectue le chiffrement du message "bring out your dead" par l'algorithme de Vigenère avec la clef "grail" :

b	r	i	n	g		o	u	t		y	o	u	r		d	e	a	d
1	17	8	13	6		14	20	19		24	14	20	17		3	4	0	3
g	r	a	i	l														
6	17	0	8	11	6	17	0	8	11	6	17	0	8	11	6	17	0	8	11
7	8	8	21	17		5	20	1		4	5	20	25		9	21	0	11
h	i	i	v	r		f	u	b		e	f	u	z		j	v	a	l

Le message chiffré "hiivr fub efuz jval" s'obtient en utilisant successivement les décalages donnés par la clef.

1. Créer un programme Python permettant de réaliser le chiffrement de Vigenère d'un fichier texte, puis créer un programme Python permettant de réaliser le déchiffrement.
2. Généraliser les programmes précédents à l'alphabet composé des 128 premiers caractères ascii.

Réponses

2)
```
from tkinter import*
# récupération du nom du fichier texte au moyen d'une boîte de dialogue
fenetre=Tk()
nomfichier=filedialog.askopenfilename(filetypes=[("fichier texte",".txt")])
fenetre.destroy()
# lecture du fichier
fichier=open(nomfichier,'r')
chaine=fichier.read()
fichier.close()
# chiffrement ROT13
CHAINE=""
for k in range(0,len(chaine)):
    if 97<=ord(chaine[k])<=122:
        CHAINE=CHAINE+chr(97+(ord(chaine[k])-84)%26)
    else:
        CHAINE=CHAINE+chaine[k]
# écriture dans le fichier
fichier=open(nomfichier,'w')
fichier.write(CHAINE)
fichier.close()
```

3)
```
from tkinter import*
# récupération du nom du fichier texte au moyen d'une boîte de dialogue
fenetre=Tk()
nomfichier=filedialog.askopenfilename(filetypes=[("fichier texte",".txt")])
fenetre.destroy()
# lecture du fichier
fichier=open(nomfichier,'r')
chaine=fichier.read()
fichier.close()
# récupération de la clef de chiffrement
clef=str(input("Clef de chiffrement=?"))
# chiffrement de Vigenère
CHAINE=""
for k in range(0,len(chaine)):
    if 97<=ord(chaine[k])<=122:
        decalage=ord(clef[k%len(clef)])-97
        CHAINE=CHAINE+chr(97+(ord(chaine[k])-97+decalage)%26)
    else:
        CHAINE=CHAINE+chaine[k]
# écriture dans le fichier
fichier=open(nomfichier,'w')
fichier.write(CHAINE)
fichier.close()
```

pour le déchiffrement, on modifie l'instruction :

```
        CHAINE=CHAINE+chr(97+(ord(chaine[k])-97-decalage)%26)
```

pour la généralisation :

```
# chiffrement de Vigenère généralisé
CHAINE=""
for k in range(0,len(chaine)):
    decalage=ord(clef[k%len(clef)])
    CHAINE=CHAINE+chr((ord(chaine[k])+decalage)%128)
```

et :

```
    CHAINE=CHAINE+chr((ord(chaine[k])-decalage)%128)
```