COURS DU CRYPTO-SYSTÈME
Série de TD/TP N°1

On rappelle qu'on a la numérotation des lettres de l'alphabet suivante :

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Exercice 1.

1. Coder le message *"la rencontre est prévue à la cafétéria"* à l'aide du chiffrement par décalage et de la clé K = 5.
2. Décoder le message *"RGNEIDVGPEWXTRAPHHXFJT"* sachant qu'il a été créé par un chiffrement par décalage.

Exercice 2.

L'analyse des fréquences d'apparition des lettres dans un message codé montre que ceux sont les lettres *K* et *O* les plus fréquentes dans ce message. Dans un texte en français les lettres les plus fréquentes sont le *A* (8.4%) et le *E* (17.26%). Sachant que le message est en français, codé en utilisant le chiffrement par décalage sur les 26 lettres de l'alphabet, déterminer la clef et déchiffrer le début du message :

SVOXFYIKNKXCVKVSQEBSOKMRODOBNOCCYVNKDC

Exercice 3.

1. Coder le message *"la rencontre est prévue à la cafétéria"* à l'aide du chiffrement par substitution et de la clé suivante :

A	B	C	D	E	F	G	H	I	J	K	L	M
X	N	Y	A	H	P	O	G	Z	Q	W	B	T
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
S	F	L	R	C	V	M	U	E	K	J	D	I

2. Est-il possible de décoder le message "YHVMQUVMH" codé par un chiffrement par substitution sans connaître la clé. Décoder ce message sachant qu'il a été créé avec la clé précédente.

Exercice 4.

1. Définir le chiffrement de Hill.

2. Coder le message *"la rencontre est prévue à la cafétéria"* à l'aide de cette méthode et de la clé suivante.

$$K = \begin{pmatrix} 1 & 2 & 3 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

3. Décoder le message *"CJLRYMLPQUNETBQJJHME"* sachant qu'il a été créé avec la clé précédente.

Exercice 5.

On suppose que le message *"c'est fini"* est codé par la méthode de Hill avec $m = 2$. On obtient le message "UYSJVPZL". Déterminer la clé du codage.

Exercice 6.

Une transposition simple opérant par remplissage d'un rectangle en lignes et relèvement par colonnes fonctionne comme suit : la clé est une suite de lettres, mot ou phrase, comme par exemple A , Chacune des lettres de la clé est numérotée, à partir de ECRITURE. et suivant l'ordre alphabétique. Sur notre exemple, cela donne :

E	C	R	I	T	U	R	E
2	1	5	4	7	8	6	3

Lors du chiffrement, le texte clair est écrit sur des lignes de même longueur que la clé, ces lignes étant disposées l'une au-dessus de l'autre pour former un rectangle :

E	C	R	I	T	U	R	E
2	1	5	4	7	8	6	3
R	A	Y	M	O	N	D	Q
U	E	N	E	A	U	E	S
T	U	N	A	U	T	E	U
R	F	A	N	T	A	S	T
I	Q	U	E				

On relève ensuite les colonnes dans l'ordre déterminé par les nombres associés aux lettres de la clé : AEUFQ RUTRI QSUTM EANEY NNAUD EESOA UTNUT A

1. Le cryptogramme suivant :

PVSNO	UPNOR	AYREA	VDNLQ	SNDEE
AUEUC	AEUEI	TOLDG	EEENU	EAEMS
ALLAA	UNFDE	LNUCL	ETGED	UITLN
LIEMC	QEERE	IE		

   a été construit suivant ce procédé avec le mot-clé *"QUENEAU"*. Retrouvez le texte clair.

2. Que pourrait-on faire si on ne disposait pas de la clé, mais seulement de sa longueur ? Et si on ne savait rien de la clé ?

TP (*Chiffrement de Vigenère*)

Objectif : Programmer la méthode de chiffrement Vigenière et savoir comment Cryptanalyser un cryptogramme.

Au XVI e siècle, Blaise de Vigenère a modernisé le codage de César, très peu résistant, de la manière suivante. Au lieu de décaler toutes les lettres du texte de la même manière, on utilise un texte clé qui donne une suite de décalages. Prenons par exemple la clé concours . Pour crypter un texte, on code la première lettre en utilisant le décalage qui renvoie a sur c (la première lettre de la clé). Pour la deuxième lettre, on prend le décalage qui envoie le a sur le o (deuxième lettre de la clé) et ainsi de suite. Pour la neuvième lettre du texte (la clé en contient que huit) on reprend à partir de la première lettre de la clé. Sur l'exemple genieinformatique avec la clé examen , on obtient :

g	e	n	i	e	i	n	f	o	r	m	a	t	i	q	u	e

La quatrième ligne donne le code de la clé (décalage à appliquer), la cinquième ligne donne la somme de la deuxième et la quatrième (modulo 26), la sixième ligne donne le texte crypté.

Travail à effectuer :

1. Ecrire une fonction ***codageVigenere*** qui prend en paramètres une liste représentant le texte à crypter et une liste d'entiers donnant la clé servant au codage, et qui retourne une liste contenant le texte crypté.
   Afin d'essayer de deviner le texte original sans la clé (de « cracker » le code), on procède en deux temps. D'abord on détermine la longueur k de la clé c, ensuite on détermine les lettres composant c.
   La première étape est la plus difficile. On remarque que deux lettres identiques dans t, le texte crypté, espacées de $l \times k$ caractères (l entier) sont codées par la même lettre dans t. Cette condition n'est passuffisante pour déterminer la longueur k de la clé c car des répétitions peuvent apparaître dans t sans qu'elles existent dans le texte original (les trois g consécutifs dans l'exemple ont pour origine trois lettres différentes). De plus, la même lettre dans le texte original peut être cryptée différemment dans t.
   Nous allons rechercher des répétitions non pas d'une lettre mais de séquences de lettres dans t puisque deux séquences de lettres répétées dans le texte original, dont les premières lettres sont espacées par $l \times l$ caractères sont aussi cryptées par deux mêmes séquences dans t.
   Dans la suite de l'énoncé, on ne considère que des séquences de taille 3 en supposant que toute répétition d'une séquence de trois lettres dans t provient exclusivement d'une séquence de trois lettres répétées dansle texte original. Ainsi, la distance séparant ces répétitions donne des multiples de k (i.e. les entiers l). La valeur de k est obtenue en prenant le PGCD de tous ces multiples. Si le nombre de répétitions est suffisant (i.e. si le texte est assez long), on a de bonnes chances d'obtenir la valeur de k. On suppose donc que cette assertion est vraie.

2. Ecrire une fonction ***pgcd*** qui calcule le PGCD de deux entiers positifs ou nuls passés en paramètres, et ce par soustractions successives.
3. Ecrire une fonction ***pgcdDistancesEntreRepetitions*** qui prend en paramètres le texte crypté t de longueur n et un entier $i \in [0, n-3]$ qui est l'indice d'une lettre dans t. La fonction retourne le PGCD de toutes les distances entre les répétitions de la séquence de 3 lettres $t[i]$, $t[i+1]$, $t[i+2]$ dans le texte $t[i+3]\ t[i+4] \ldots t[n-1]$. Cette fonction retourne 0 s'il n'y a pas de répétition.
4. Ecrire une fonction ***longueurCle*** qui prend en paramètres le texte crypté t et qui retourne la longueur k probable de la clé de codage.
5. Donner le nombre maximal d'opérations réalisées par la fonction ***longueurCle*** en fonction de la longueur de t. On ne comptera que le nombre d'appels à la fonction ***pgcd***.
6. Une fois la longueur de la clé connue, donnez une idée d'algorithme permettant de retrouver chacune des lettres de la clé. Il s'agit de décrire rapidement l'algorithme et non d'écrire le programme.
7. *Question bonus*. Ecrire une fonction ***decodageVigenereAuto*** qui prend en paramètres le texte crypté t et qui retourne le texte original probable.