

■ T.P. 7 ■

Chiffre de Vigenere

En 1586, Blaise de Vigenère, publia un *traité sur les chiffres* dans lequel il propose une modernisation du chiffre de César. 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 chiffrer un texte, on chiffre la première lettre en utilisant le décalage qui envoie le **a** sur le **c** (la première lettre de la clé). Pour la deuxième lettre, on prend le décalage qui envoie le **a** sur le **o** (la seconde lettre de la clé) et ainsi de suite. Pour la huitième lettre, on utilise le décalage de **a** vers **s**, puis, pour la neuvième, on reprend la clé à partir de sa première lettre. Sur l'exemple **ecolepolytechnique** avec la clé **concours**, on obtient (la première ligne donne la clé, la deuxième le message à chiffrer et la troisième le message chiffré) :

c	o	n	c	o	u	r	s	c	o	n	c	o	u	r	s	c	o
e	c	o	l	e	p	o	l	y	t	e	c	h	n	i	q	u	e
g	q	b	n	s	j	f	d	a	h	r	e	v	h	z	i	w	s

On suppose que les caractères utilisés sont uniquement les minuscules de l'alphabet latin.

1. Donner le chiffre du texte **becunfromage** en utilisant la clé de chiffrement **jean**.

2. Écrire la fonction `chiffrement_vigenere(t, c)` qui prend comme arguments une liste **t** représentant le texte à chiffrer, et une liste **c** représentant la clé servant au chiffrement ; et qui retourne une liste contenant le texte chiffré.

On suppose disposer d'un texte *tc* assez long, chiffré par la méthode de Vigenère, et on veut retrouver le texte *t* d'origine. Pour cela, on doit trouver la clé *c* ayant servi au chiffage. On procède en deux temps :

- (i) Déterminer la longueur *k* de la clé *c*.
- (ii) Déterminer les lettres composant *c*.

La première étape est la plus difficile. On remarque que deux lettres identiques dans *t* espacées de $\ell \times k$ caractères (où ℓ est un entier et *k* est la longueur de la clé) sont chiffrées par la même lettre dans le texte chiffré *tc*. Mais cette condition n'est pas suffisante pour déterminer la longueur *k* de la clé *c* puisque des répétitions peuvent apparaître dans *tc* sans qu'elles existent dans *t*. Par exemple, les lettres **t** et **n** sont toutes deux chiffrées par la lettre **h** dans le texte chiffré à partir de **ecolepolytechnique** avec **concours** comme clé. Pour éviter ce problème, on recherche les répétitions non pas d'une lettre mais de séquences de lettres dans *tc* puisque deux séquences de lettres répétées dans *t* dont les premières lettres sont espacées par $\ell \times k$ caractères, sont aussi chiffrées par deux mêmes séquences dans *tc*.

Dans la suite, on ne considère que des séquences de longueur 3 en supposant que toute répétition d'une séquence de 3 lettres dans *tc* provient exclusivement d'une séquence de 3 lettres répétée dans *t*. Ainsi, la distance séparant ces répétitions donne des multiples de *k*. La valeur de *k* est alors obtenue en prenant le plus grand commun diviseur de tous ces multiples. Si le nombre de répétitions est suffisant, on a de bonnes chances d'obtenir la valeur de *k*. On suppose donc que cette assertion est vraie.

3. Écrire la fonction `pgcd_distances_entre_repetitions(tc, i)` qui prend en argument le texte chiffré **tc** de longueur *n* et un entier $i \in \llbracket 0, n-3 \rrbracket$ qui est l'indice d'une lettre de **tc** et qui retourne le plus grand commun diviseur de toutes les distances entre les répétitions de la séquence de 3 lettres **[tc[i], tc[i + 1], tc[i + 2]]** dans la suite du texte **[tc[i + 3], tc[i + 4], ..., tc[n - 1]]**. Cette fonction retourne 0 s'il n'y a pas de répétition.

4. Écrire la fonction `longueur_cle(tc)` qui prend en argument le texte chiffré **tc** et qui retourne la longueur *k* de la clé de chiffrement.

5. Écrire la fonction `calcul_cle(tc, k)` qui renvoie la clé de chiffage connaissant le texte chiffré **tc** et la longueur **k** de la clé.

6. Écrire la fonction `dechiffrement_vigenere(tc)` qui prend en argument la liste **tc** représentant le texte chiffré et qui retourne le texte d'origine.

*Vous pourrez utiliser le fichier **tp7_mystere.txt** pour tester votre programme.*