TD6 : Algorithme, invariant de boucle et complexité (2/2)

Exercice 1 : Algorithme de chiffrement d'une chaîne de caractères

Considérons l'algorithme de chiffrement de chaînes de caractères dit « de Vigenère ». Il consiste à « additionner » les caractères du texte à chiffrer avec ceux d'une clé de chiffrement. Par exemple, le chiffrement de la chaîne « Cherchez au pied de l'arbre » avec la clé « indice » peut s'illustrer ainsi :

- on place la clé en regard du texte à chiffrer, en répétant la clé autant de fois que nécessaire pour couvrir le texte, et en ignorant les caractères qui ne sont pas des lettres (ils seront laissés inchangés par l'algorithme de chiffrement) :

C	h	e	r	c	h	e	z		a	u		p	i	e	d		d	e		l	'	a	r	b	r	e
i	n	d	i	c	e	i	n		d	i		c	e	i	n		d	i		c	'	e	i	n	d	i

- on remplace chaque lettre de la clé par sa position dans l'alphabet (0 pour 'a', 1 pour 'b'…),

C	h	e	r	c	h	e	z		a	u		p	i	e	d		d	e		l	'	a	r	b	r	e
8	13	3	8	2	4	8	13		3	8		2	4	8	13		3	8		2	'	4	8	13	3	8

- on remplace chaque lettre du texte à chiffrer par la lettre située *d* positions plus loin dans l'alphabet, où *d* est le nombre indiqué par la clé (si on dépasse z, on reboucle sur a, b etc.) :

C	h	e	r	c	h	e	z		a	u		p	i	e	d		d	e		l	'	a	r	b	r	e
K	u	h	z	e	l	m	m		d	c		r	m	m	q		g	m		n	'	e	z	o	u	m

Notez qu'une lettre identique dans le texte à chiffrer ne produit pas forcément le même code (ex. le premier et deuxième e de Cherchez donnent respectivement un h et un m). Et notez qu'un même code n'est pas forcément issu d'une même lettre (ex. les deux m successifs du troisième mot sont issus d'un i et d'un e). Cela rend le processus de décryptage très difficile sans la clé.

a. Si *x* est le code ASCII d'une lettre à chiffrer et *y* le code ASCII de la lettre de la clé située en regard, quelle formule permet de calculer le code ASCII résultant ? On supposera pour cette question que les deux lettres sont des minuscules. Indice : que vaut *y* – 'a' ?

b. Ecrire en langage algorithmique la procédure de chiffrement, dont voici l'entête :

> **Procédure** chiffrer (texte : chaîne de caractères, cle : chaîne de caractères, result : chaîne de caractères)
> **Préconditions** : texte contient un ou plusieurs caractères suivis d'un caractère de terminaison '\0'. cle contient une ou plusieurs lettres minuscules suivie de '\0'. result est une chaîne déjà allouée en mémoire, assez grande pour contenir le texte crypté (incluant la place pour le caractère de terminaison).
> **Postcondition** : result contient la version cryptée de texte. Seules les lettres (majuscules ou minuscules) non accentuées sont cryptées, les autres caractères sont recopiés tels quels.
> **Paramètres en mode donnée** : texte, cle
> **Paramètre en mode donnée-résultat** : result

c. Supposons que l'on veuille chiffrer, avec une clé de longueur k, une chaîne constituée exclusivement de L minuscules, avec L > k. Comptez le nombre d'opérations de chaque type (affectations d'entiers, comparaison de caractères, etc.) pour évaluer la complexité en temps de cet algorithme de chiffrement. Qui de L ou de k influence le plus le temps d'exécution ? Ce temps sera-t-il plus long ou plus court avec une plus longue clé ?

d. Quelle serait l'entête de la procédure en C++ ?

e. Donner un invariant de la boucle introduite dans la procédure chiffrer.