Arithmétique TI-Nspire	Cryptographie, chiffrement affine

Objectifs

La charmante Alice souhaite envoyer le message suivant à Bob – une phrase de Cantor – en utilisant un chiffrement affine :

L'essence des mathématiques, c'est la liberté.

Première étape : codage des lettres par leur rang dans l'alphabet

En utilisant leur rang dans l'alphabet, on peut naturellement associer à chaque lettre un nombre entier, selon le tableau suivant :

a	b	c	d	e	f	g	h	i	j	k	l	m
00	01	02	03	04	05	06	07	08	09	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

1) On demande le codage de *L'essence* ou *lessence* puisqu'Alice ne se soucie ni des espaces, ni des apostrophes, ni des majuscules. Le tableau nous donne la correspondance sans difficulté :

1104181804130204

2) a) Observons tout d'abord que les lettres minuscules de l'alphabet sont codées de 97 pour "a" à 122 pour "z". Il suffit donc de retirer 97 à **ord** pour obtenir un codage qui varie de 0 à 25, comme celui qui est demandé.

On peut donc saisir dans la cellule B1 l'instruction **=ord(A1)–97**. Il suffit ensuite de recopier vers le bas, autant de fois que nécessaire.

Il est indispensable que les lettres de la colonne A aient été saisies entre guillemets, pour être traitées comme des chaines de caractères et non des variables.

b) On obtient exactement le même résultat en saisissant dans la zone grisée de la colonne B : **ord(clair)–97** ou **ord(A[])–97**, **clair** étant le nom donné à la colonne A (le nom par défaut est **A[]**)

Deuxième étape : le chiffrement affine en action

Nous supposerons dans la suite que $a = 3$ et $b = 7$. On peut placer ces valeurs dans les cellules H1 et H2 et les mémoriser dans les variables **a** et **b**.

1) Pour chiffrer la première lettre du message, on part du rang de la lettre "l" dans l'alphabet, soit 11. On calcule ensuite le reste dans la division euclidienne de $a \times 11 + b = 3 \times 11 + 7 = 40$ par 26 : cela donne 14, qui correspond à la lettre "o".

Pour chiffrer la deuxième lettre du message, "e" de rang 4, on calcule le reste dans la division euclidienne de $a \times 4 + b = 3 \times 4 + 7 = 19$ par 26 : cela donne 19, qui correspond à la lettre "t".

2) a) Poursuivons dans la feuille de calcul précédemment ouverte.

La colonne C effectue le chiffrement affine proprement dit : en C1, on saisit **=mod(a·B1+b,26)** et on recopie vers le bas.

Enfin la colonne D associe à chaque nombre obtenu en C la lettre qui lui correspond, avec en D1, l'instruction **=char(C1+97)** recopié vers le bas.

Le message codé apparaît dans la colonne D, nommée **chif_affine**.

© Texas Instruments 2012 / Photocopie autorisée

	A	B	C chif_affine	D	E	F	G	H	I	J	K
◆											
1	l	11	14	o			a=	**3**			
2	e	4	19	t			b=	**7**			
3	s	18	9	j							
4	s	18	9	j							
5	e	4	19	t							
6	n	13	20	u							
7	c	2	13	n							
8	e	4	19	t							
9	d	3	16	q							
10	e	4	19	t							
11	s	18	9	j							
12	m	12	17	r							
13	a	0	7	h							
14	t	19	12	m							
15	h	7	2	c							
16	e	4	19	t							
17	m	12	17	r							
18	a	0	7	h							

J4

b) Dans la zone grisée de la colonne D, on écrit =char(c[]+97).

3) Le chiffrement proposée consiste à faire $a = 1$ et $b = 3$: le "a" est alors bien codé par un "d". Le chiffrement de ce message est alors immédiat.

	A	B	C chif_affine	D	E	F	G	H	I	J	K
◆											
1	v	21	24	y			a=	**1**			
2	e	4	7	h			b=	**3**			
3	n	13	16	q							
4	i	8	11	l							
5	v	21	24	y							
6	i	8	11	l							
7	d	3	6	g							
8	v	21	24	y							
9	i	8	11	l							
10	c	2	5	f							
11	i	8	11	l							

© Texas Instruments 2012 / Photocopie autorisée

Troisième étape : l'écriture d'une fonction

1) mes contient le message que l'on veut coder ; **a** et **b** sont les deux paramètres du chiffrement affine.

```
* chiaf                                                    7/8
Define chiaf(mes,a,b)=
Func
Local
dim(mes)→n:"[ ]"→cod
For i,1,n
  mid(mes,i,1)→let


EndFor
Return cod
EndFunc
```

2) La première instruction donne le nombre de caractères de **mes**, la seconde met une chaîne de caractère vide dans la variable **cod**, qui recevra au fur et à mesure chaque lettre du message codé.

3) Dans la boucle **For**, on examine chaque lettre du message, on la code en chiffre selon son rang dans l'alphabet, on applique le chiffrement affine et le résultat obtenu est de nouveau transformé en lettre, lettre que l'on concatène à la variable **cod**, qui contient le message codé.

4) On peut compléter la fonction de la façon suivante :

```
  chiaf                                                    7/9
Define chiaf(mes,a,b)=
Func
Local n,cod,i,let,r0,r1
dim(mes)→n:"[ ]"→cod
For i,1,n
  mid(mes,i,1)→let
  ord(let)-97→r0
  mod(a·r0+b,26)→r1
  cod&char(r1+97)→cod
EndFor
Return cod
EndFunc
```

Si l'on chiffre le message initial d'Alice, on obtient bien la même chose que dans le tableur :

```
chiaf("lessencedesmathematiquescestlaliberte",3,7)        "otjjtuntqtjrhmctrhmfdptjntjmohofktgmt"
```

5) Choix de *a* et *b*

a) On observe que des lettres différentes sont codées de la même façon. Si à la limite on peut envisager de coder un message, l'opération inverse, le décodage, en sera complètement impossible : il faut qu'une lettre codée étant donnée, on ne puisse remonter que d'une seule façon à la lettre dont elle provient.
Mathématiquement, un codage doit réaliser une *bijection* de l'ensemble des lettres de l'alphabet sur lui-même.

b) On se rend compte, après quelques essais, que le chiffrement réalise bien une bijection de l'alphabet sur lui-même dans le cas où a est premier avec 26, b prenant une valeur quelconque.

c) Il suffit de montrer que si deux lettres chiffrées sont les mêmes, elles proviennent nécessairement de la même lettre. En termes mathématiques, nous devons montrer que si on a $al + b \equiv al' + b$ (26), alors $l \equiv l'$ (26), l et l' étant des entiers naturels compris entre 0 et 25.
Or cette équation avec congruence, $al + b \equiv al' + b$ (26), se résout facilement. Elle équivaut à :

$$a(l - l') \equiv 0 \ (26).$$

a et 26 étant premier entre eux, on sait qu'il existe u et v entiers relatifs tels que $au + 26v = 1$. Cette égalité, prise modulo 26 donne :

$$au \equiv 1 \ (26).$$

Il existe donc un entier u, nécessairement non nul, qui est en quelque sorte l'inverse de a modulo 26.
En multipliant $a(l - l') \equiv 0 \ (26)$ par u, on arrive à :

$$l \equiv l' \ (26).$$

Comme l et l' sont des entiers compris entre 0 et 25, nécessairement $l = l'$.

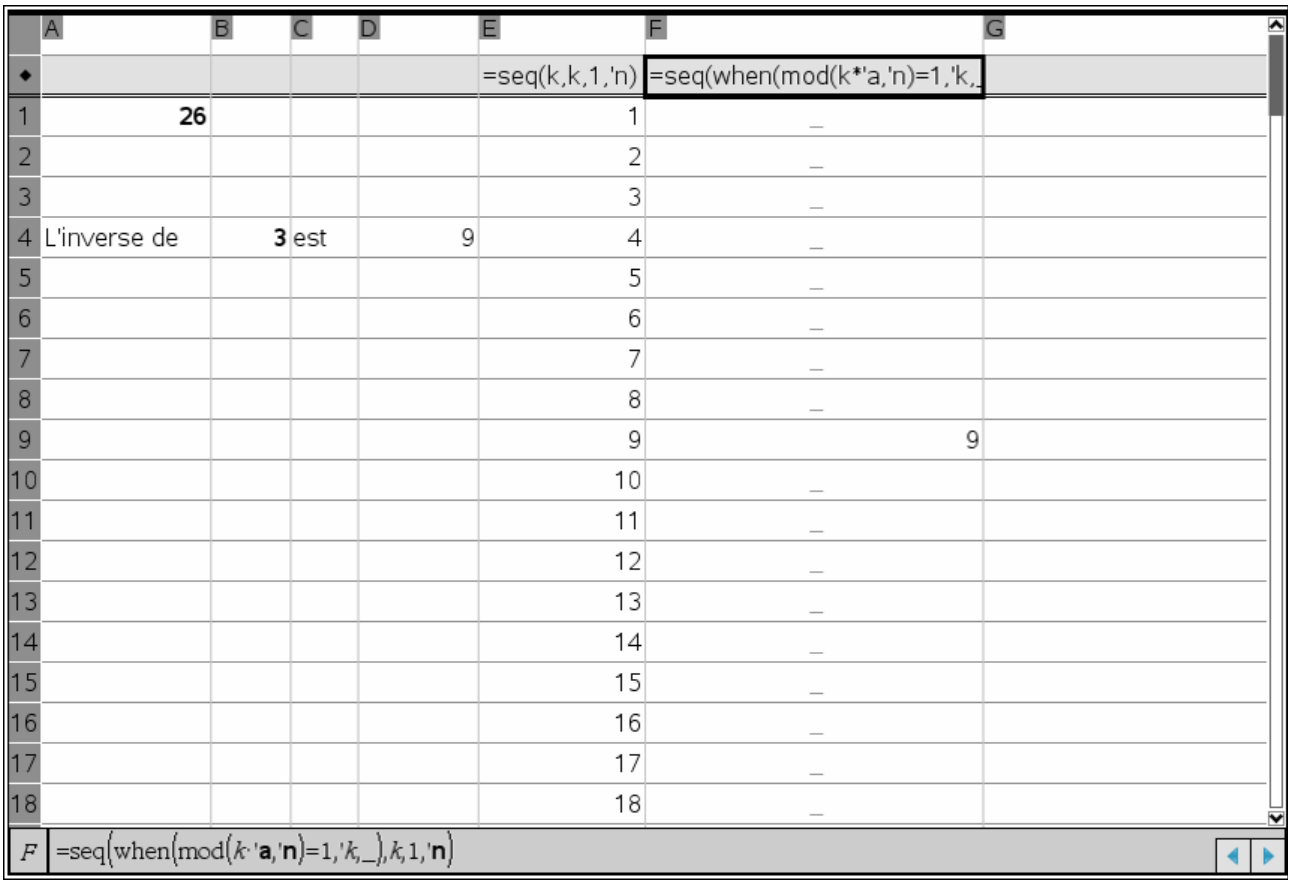
d) Répondons tout d'abord à la question de savoir combien d'entiers compris entre 0 et 25 sont premiers avec 26.
Il faut d'abord retirer tous les nombres pairs, 0 y compris, puis 13. Il reste 12 nombres compris entre 0 et 25 et premiers avec 26.
On peut alors en déduire le nombre de chiffrements affines possibles : 12 choix possibles pour a, à multiplier par les 26 valeurs possibles de b, de 0 à 25, ce qui donne un total de 312 chiffrements affines, 311 même si l'on exclut le chiffrement identique ($a = 1$ et $b = 0$)… facile à déchiffrer dès que l'on sait lire !

Quatrième étape : déchiffrement par Bob du message d'Alice

Bob reçoit donc le message chiffré précédent et connaît, c'est indispensable, les clés de chiffrement d'Alice, à savoir $a = 3$ et $b = 7$:

"otjjtuntqtjrhmctrhmfdptjntjmohofktgmt"

1) a) En effet, $3 \times 9 = 27$ est bien congru à 1 modulo 26.
Remarquons qu'une recherche systématique peut être entreprise au tableur TI-Nspire, parce que 26 est un entier de taille modeste :



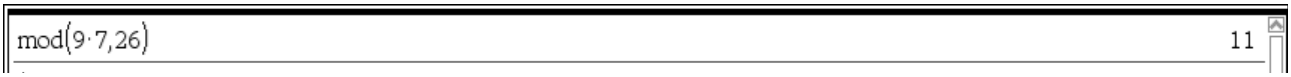
Toutes les instructions sont apparentes, sauf celle saisie en D4 qui est :

=delvoid(f[])[1].

b) On peut poursuivre la résolution de l'équation en multipliant les deux membres par l'inverse de 3 modulo 26, soit 9. On arrive à :

$$x \equiv 7 \times 9 \equiv 63 \equiv 11 \ (26).$$

© Texas Instruments 2012 / Photocopie autorisée



La première lettre du message d'Alice est donc un "l".

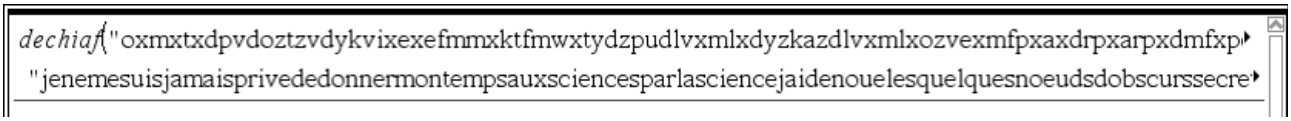
2) Des calculs analogues conduisent à $19 \equiv 3x + 7\ (26)$ car "t" est la lettre de rang 19 de l'alphabet.
Ceci équivaut à $3x \equiv 12\ (26)$ soit à $x \equiv 108\ (26)$.

La lettre est bien un "e".

3) Écriture d'une fonction dechiaf pour le déchiffrement des messages

a) La fonction que l'on écrit est la suivante :

b) Il suffit d'utiliser la fonction que l'on vient d'écrire, en utilisant les bonnes valeurs de a et b, à savoir 19 et 25 :



Une rapide recherche sur Internet, à partir du premier vers (à mettre entre guillemets dans Google), nous donne l'origine de ce texte :

> Je ne me suis jamais privé de donner mon temps aux sciences
> Par la science j'ai dénoué les quelques nœuds d'obscurs secrets
> Après soixante-douze années de réflexion sans jour de trêve
> Mon ignorance, je la sais...

C'est un quatrain – cité dans *Le théorème du perroquet* de Denis Guedj –du grand poète et mathématicien, Omar Khayyam qui a vécu de 1048 à 1131.

Un dernier rebondissement : l'infâme Charles essaie d'intercepter la correspondance d'Alice et Bob

Voici le message intercepté :
"djcbdfbgdgbxovjbgsktdzhbcbdrtsubrtsjzhbddvgsdjrkcbdxbdsktoxbzhjcdgbdtwbgsktdtzhbckvjgsctwjbbdsxvrkcjzhbb"
Plusieurs stratégies nous permettent de déchiffrer ce message même si nous n'en connaissons pas les clés.

1) La force brute

a) La fonction proposée essaie tous les codages possibles – il y en a 311 – et fait afficher le résultat obtenu.

b) Comme les codages possibles sont en nombre limité, on peut espérer assez vite retrouver le message initial.

c) Les différents codages possibles défilent très rapidement. Il reste, après, à examiner ce qu'on a obtenu. La réponse est :
"silesgensnecroientpasquelesmathematiquessontsimplescestparcequilsnesaventpasaquelpointlavieestcompliquee"

© Texas Instruments 2012 / Photocopie autorisée

soit
"si les gens ne croient pas que les mathématiques sont simples c'est parce qu'ils ne savent pas à quel point la vie est compliquée"
… citation en général attribuée à Von Neumann.

2) L'intelligence en action ?

a) On s'appuie sur la fréquence des lettres en français : dans l'ordre décroissant des fréquences, on trouve les lettres e, s, a, r, t, i, n, u, l, o, c… souvent utilisées par l'Oulipo[1].

Dans le message qu'on cherche à déchiffrer, les lettres les plus fréquentes sont dans cet ordre b (18 occurrences), d (14 occurrences), j, s et t (8 occurrences chacune).
On peut donc penser que le "e" a été traduit par "b" et le "s" par "d".
On en déduit les congruences suivantes, modulo 26

$$\begin{cases} 4a+b \equiv 1 \\ 18a+b \equiv 3 \end{cases}$$

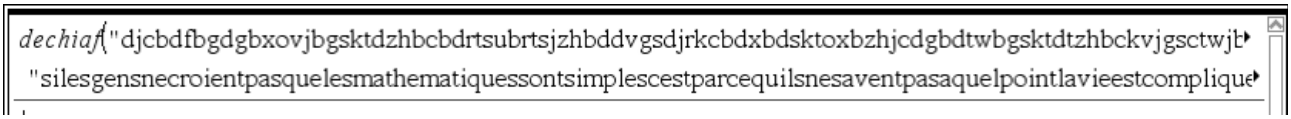
b) Par différence, on en déduit que $14a \equiv 2\ (26)$, soit $14a \equiv 2\ (26)$ soit $7a \equiv 1\ (13)$.

Tout revient donc à trouver l'inverse de 7 modulo 13 : c'est 2 (voir par exemple la feuille de calcul que nous avons faite plus haut).
En multipliant par 2 les deux membres de la dernière congruence, on arrive à :

$$a \equiv 2\ (13).$$

Autrement dit, a peut s'écrire sous la forme $13k + 2$; on sait, par ailleurs, que a est un entier compris entre 0 et 25, premier avec 26 ; il ne reste que $a = 15$, puisque 2 n'est pas premier avec 26.
La première congruence nous donne $b \equiv 1 - 4a \equiv 1 - 60 \equiv -59 \equiv 19\ (26)$.
La deuxième congruence est bien vérifiée car : $15 \times 18 + 19 = 289 \equiv 3\ (26)$...
Un seul couple solution apparaît, $(a,b) = (15,19)$.

c) Il reste à tester le décodage avec ces valeurs de a et de b. Ce sont bien celles qui conviennent. Sinon il faudrait tester d'autres choix pour les lettres les plus fréquentes.



[1] L'Ouvroir de Littérature Potentielle, généralement désigné par OuLiPo (ou Oulipo), est un groupe international de mathématiciens et de littéraires, fondé par le mathématicien François Le Lionnais et l'écrivain Raymond Queneau. Georges Perec en fut membre.

© Texas Instruments 2012 / Photocopie autorisée