

Devoir en classe III.2b

04-07-2003

Introduction (Cryptographie) :

Une méthode pour coder une chaîne de caractères, consiste à décaler les caractères d'un texte donné, un par un, de N positions dans l'alphabet. (Voir Exemple 1)

Pour que le texte puisse être codé, il doit être en lettres majuscules et ne plus contenir de ponctuations - il faut qu'il soit 'pure'.

Le nombre de positions à décaler est recalculé à l'aide de la fonction modulo lorsque qu'il est plus grand que 26, (ceci est nécessaire à ce que le caractère en question ne soit décalé dans l'alphabet que de $N = 26$ positions au maximum). (Voir Exemple 2)

Un décalage de 0 ou de 26 positions n'est pas un décalage. (Voir Exemple 3)

66	67	68	69	70	71	72	73	74	ASCII	
A	B	C	D	E	F	G	H	I	J	
01	02	03	04	05	06	07	08	09	10	position

75	76	77	78	79	80	81	82	83	84	ASCII
K	L	M	N	O	P	Q	R	S	T	
11	12	13	14	15	16	17	18	19	20	position

85	86	87	88	89	90	ASCII
U	V	W	X	Y	Z	
21		23	24	25	26	position

Exemple 1 ($N = 1$)

S	E	H	R	G	E	E	H	R	T	E	D	A	M	E	N	U	N	D	H	E	R	R	E	N
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
T	F	I	S	H	F	F	I	S	U	F	E	B	N	F	O	V	O	E	I	F	S	S	F	O

Exemple 2 ($N = 40 \text{ MOD } 26 = 14$)

S	E	H	R	G	E	E	H	R	T	E	D	A	M	E	N	U	N	D	H	E	R	R	E	N
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
G	S	V	F	U	S	S	V	F	H	S	R	O	A	S	B	I	B	R	V	S	F	F	S	B

Exemple 3 ($N = 26 \text{ MOD } 26 = 0$)

S	E	H	R	G	E	E	H	R	T	E	D	A	M	E	N	U	N	D	H	E	R	R	E	N
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
S	E	H	R	G	E	E	H	R	T	E	D	A	M	E	N	U	N	D	H	E	R	R	E	N

Devoir en classe III.2b

04-07-2003

1. **(10p.)** Développez le sous-programme `RENDRE_PURE` lequel retourne une chaîne de caractères. Cette chaîne est dite 'pure' ce qui veut dire, qu'elle ne contient plus de ponctuations (voir exemples d'introduction). La chaîne de caractères qui est à analyser, est transmis par valeur. La fonction `CAR_PONC(C:char):boolean` peut être utilisée. Elle retourne 'vrai' si le caractère passé par valeur est un caractère de ponctuation ou un signe de ponctuation.

2. **(8p.)** Développez la fonction `MAJSTR`. La valeur de retour est une chaîne de caractères. Elle est basée sur une chaîne de caractères passée par valeur. La fonction transforme tout caractère minuscule en caractère majuscule. (Astuce : `ORD('Z')=90,ORD('a')=97.`)
Exemple : `MAJSTR('AbCd')` = 'ABCD'

3. **(8p.)** Développez le sous-programme `STR2TAB` qui inscrit une chaîne de caractères dans un tableau du type `tTAB`.
`type tTAB = array [1..MAX_DIM,1..256] of char`
Les changements faits dans le tableau sont nécessaires dans la suite du programme.
La chaîne, le tableau ainsi qu'un indice sont transmis au sous-programme. L'indice sert à identifier la position du tableau, où la chaîne est à inscrire.
Attention : Si l'indice transmis est plus grand que `MAX_DIM`, `STR2TAB` retourne `FALSE` - si l'indice est valide, `STR2TAB` retourne `TRUE`.

4. **(5p.)** Développez la fonction `CRYPTCHAR` qui retourne un caractère codé, basé sur le caractère passé par valeur. Le nombre N (= positions à décaler) est transmis par valeur. Le codage se fait selon le principe suivant :

Système d'encodage :
 $P = \text{ORD}(\text{caractère}) - 64$ est la position des caractères dans l'alphabet normal;
 $Q = P + N$ donne la nouvelle position du caractère dans l'alphabet décalé, en sachant que si $N > 26$ alors $N = N \bmod 26$.
Si le résultat $Q > 26$, alors il faut calculer la nouvelle position
 $Q = Q \bmod 26$. Le caractère codé devient : `CHR(Q+64)`.

5. **(4p.)** Développez le sous-programme `CRYPTSTR` qui retourne la version codée d'une chaîne de caractères passée par valeur. Le nombre N (= positions à décaler) est transmis par valeur. Utilisez le sous-programme `CRYPTCHAR` !

Devoir en classe III.2b

04-07-2003

6. **(13p.)** Développez le sous-programme `ALPHA2TAB` lequel inscrit l'alphabet '*normal*' (limité aux 26 caractères majuscules) ainsi que l'alphabet '*décalé*' décalé de N positions à l'aide du sous-programme `STR2TAB`.
Le tableau est transmis au sous-programme. L'alphabet *normal* doit se trouver à la première position du tableau, l'alphabet *décalé* (= codé) à la deuxième position.
`ALPHA2TAB` retourne `TRUE` si l'opération a pu se faire, et `FALSE` le cas échéant.
7. **(12p.)** Développez la procédure `DECRYPT`, qui transforme une chaîne codée en chaîne décodée, en utilisant les deux alphabets (*normal* et *décalé*) qui ont été enregistrés dans un tableau à l'aide du sous-programme `ALPHA2TAB`.
Les paramètres transmis au sous-programme sont la chaîne de caractères à décoder et le tableau du type `tTAB`.
On suppose que la chaîne codée est valide - elle est donc décodable !

-FIN-