Chiffrement de César

Guillaume CONNAN

Lycée Jean PERRIN

Sommaire


VA A ROME PORTER UN MESSAGE À CÉSAR. TU LUI DIRAS : " TOUTE LA GAULE EST OCCUPÉE ", IL TE DEMANDERA : " TOUTE ? ", TU LUI RÉPONDRAS : " TOUTE ! " IL COMPRENDRA.
Uderzo & Goscinny

Comme le disait Suétone (70-127) dans La vie des 12 Césars :

> *Extant et ad Ciceronem, item ad familiares domesticis de rebus, in quibus, si qua occultius perferenda erant, per notas scripsit, id est sic structo litterarum ordine, ut nullum uerbum effici posset : quae si qui inuestigare et persequi uelit, quartam elementorum litteram, id est D pro A et perinde reliquas commutet.*

Ce que César aurait peut-être écrit sous cette forme :

Rkgnag rg nq Pvprebarz, vgrz nq snzvyvnerf qbzrfgvpvf qr erohf, va dhvohf, fv dhn bpphygvhf cresreraqn renag, cre abgnf fpevcfvg, vq rfg fvp fgehpgb yvggrenehz beqvar, hg ahyyhz hreohz rssvpv cbffrg : dhnr fv dhv vahrfgvtner rg crefrdhv hryvg, dhnegnz ryrzragbehz yvggrenz, vq rfg Q ceb N rg crevaqr eryvdhnf pbzzhgrg.

Certains seront sûrement plus à l'aise avec cette nouvelle transcription du même texte :

> *On possède enfin de César des lettres à Cicéron, et sa correspondance avec ses amis sur ses affaires domestiques. Il y employait, pour les choses tout à fait secrètes, une espèce de chiffre qui en rendait le sens inintelligible (les lettres étant disposées de manière à ne pouvoir jamais former un mot), et qui consistait, je le dis pour ceux qui voudront les déchiffrer, à changer le rang des lettres dans l'alphabet, en écrivant la quatrième pour la première, c'est-à-dire le D pour l'A, et ainsi de suite.*

Sommaire

Vous voulez transmettre cet important message :

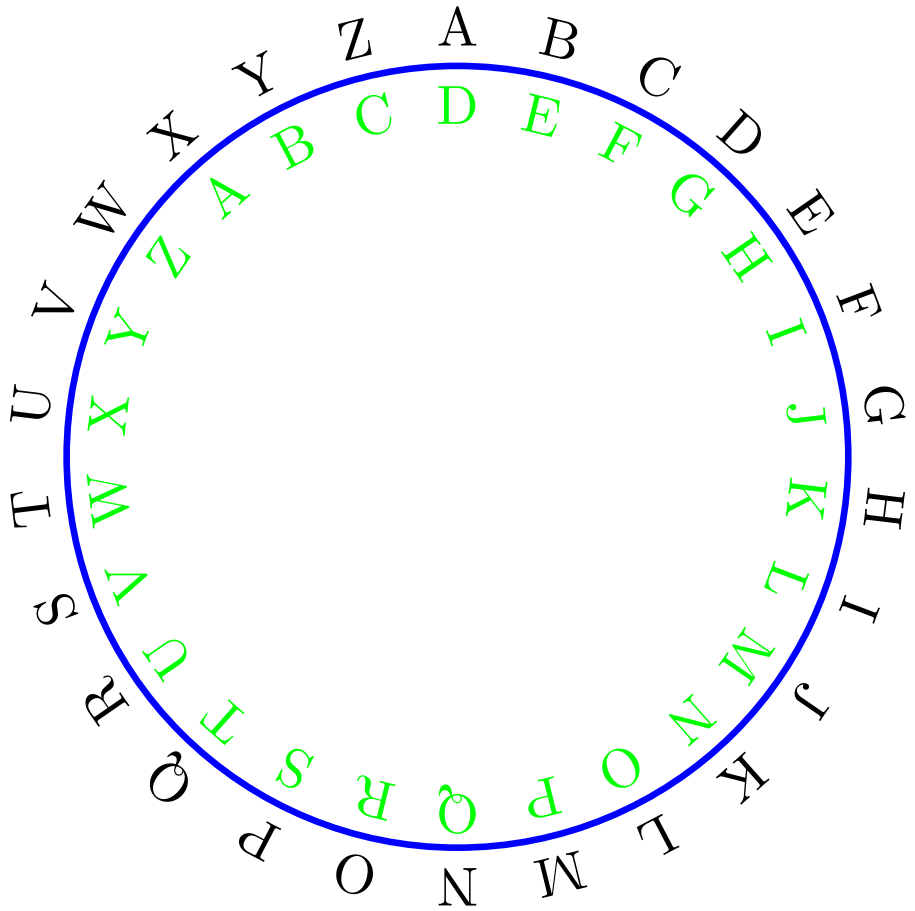
Les sanglots longs des violons de l'automne

Sommaire

Vous voulez traduire ce message intercepté par vos espions :

*eohvvhqw#prq#frhxu#g*xqh#odqjxhxu#prqrwrqh*

Que vous inspire ce dessin :



Codons chaque lettre par un nombre : $A \longmapsto 0$, $B \longmapsto 1$, etc.

Comment modéliser la chiffrement de César ?

Y a-t-il un problème ?

Qu'ont en commun 29 et 3 ?

Codons chaque lettre par un nombre : $A \longmapsto 0$, $B \longmapsto 1$, etc.
Comment modéliser la chiffrement de César ?
Y a-t-il un problème ?
Qu'ont en commun 29 et 3 ?

Codons chaque lettre par un nombre : $A \longmapsto 0$, $B \longmapsto 1$, etc.
Comment modéliser la chiffrement de César ?
Y a-t-il un problème ?
Qu'ont en commun 29 et 3 ?

Codons chaque lettre par un nombre : $A \longmapsto 0$, $B \longmapsto 1$, etc.
Comment modéliser la chiffrement de César ?
Y a-t-il un problème ?
Qu'ont en commun 29 et 3 ?

Sommaire

Supposons que nous disposions de cette table de codage

0	1 !	2 "	3 #	4 $	5 %	6 &	7 '	8 (	9)
10 *	11 +	12 ,	13 -	14 .	15 /	16 0	17 1	18 2	19 3
20 4	21 5	22 6	23 7	24 8	25 9	26 :	27 ;	28 <	29 =
30 >	31 ?	32	33 A	34 B	35 C	36 D	37 E	38 F	39 G
40 H	41 I	42 J	43 K	44 L	45 M	46 N	47 O	48 P	49 Q
50 R	51 S	52 T	53 U	54 V	55 W	56 X	57 Y	58 Z	59 [
60 \	61]	62 ^	63 _	64 ‘	65 a	66 b	67 c	68 d	69 e
70 f	71 g	72 h	73 i	74 j	75 k	76 l	77 m	78 n	79 o
80 p	81 q	82 r	83 s	84 t	85 u	86 v	87 w	88 x	89 y
90 z	91 {	92 |	93 }	94 ~	95 ê	96 ù	97 ç	98 à	99 è
100 é	101 Ê	102 Ç	103 À	104 É	105 È				

Et supposons que nous disposions d'une procédure code(“message”) qui transforme un message en une liste de codes.

Par exemple *MOI* devient

[45,47,41]

et de la fonction decode associée.

Et supposons que nous disposions d'une procédure code(“message”) qui transforme un message en une liste de codes.
Par exemple *MOI* devient
[45,47,41]
et de la fonction decode associée.

Et supposons que nous disposions d'une procédure `code("message")` qui transforme un message en une liste de codes.
Par exemple *MOI* devient
[45,47,41]
et de la fonction decode associée.

Et supposons que nous disposions d'une procédure `code("message")` qui transforme un message en une liste de codes.
Par exemple *MOI* devient
[45,47,41]
et de la fonction `decode` associée.

Tout langage de programmation de calcul programmable admet

- une fonction calculant le reste de la division euclidienne de deux entiers ;
- une fonction ajoutant un opérande à une liste

Imaginez alors un algorithme de codage selon la méthode du glorieux César.

Tout langage de programmation de calcul programmable admet

- une fonction calculant le reste de la division euclidienne de deux entiers ;
- une fonction ajoutant un opérande à une liste

Imaginez alors un algorithme de codage selon la méthode du glorieux César.

Tout langage de programmation de calcul programmable admet

- une fonction calculant le reste de la division euclidienne de deux entiers ;
- une fonction ajoutant un opérande à une liste

Imaginez alors un algorithme de codage selon la méthode du glorieux César.

Tout langage de programmation de calcul programmable admet

- une fonction calculant le reste de la division euclidienne de deux entiers ;
- une fonction ajoutant un opérande à une liste

Imaginez alors un algorithme de codage selon la méthode du glorieux César.

Sommaire

- `irem(a,b)` calcule le reste de la division euclidienne de a par b
- `concat(C,op)` ajoute l'élément op au bout de la chaîne `C`

- `irem(a,b)` calcule le reste de la division euclidienne de a par b
- `concat(C,op)` ajoute l'élément `op` au bout de la chaîne `C`

Un ASCII adapté

Les américains ont mis au point le code ASCII : ils ont donc oublié de coder nos lettres accentuées...

De plus, parmi toutes les touches codées, seules celles contenant un certains nombre de caractères nous intéressent.

Un ASCII adapté

Les américains ont mis au point le code ASCII : ils ont donc oublié de coder nos lettres accentuées...
De plus, parmi toutes les touches codées, seules celles contenant un certains nombre de caractères nous intéressent.

Codage

```
code:= (c)->{
if (c==''é'') return(100) ;
if (c==''è'') return(99) ;
if (c==''à'') return(98) ;
if (c==''ç'') return(97) ;
if (c==''ù'') return(96) ;
if (c==''ê'') return(95) ;
etc.
```

```
return(asc(c)-32);
```

À vous d'imaginer la procédure decode...

Codage

```
code:= (c)->{
if (c==''é'') return(100) ;
if (c==''è'') return(99) ;
if (c==''à'') return(98) ;
if (c==''ç'') return(97) ;
if (c==''ù'') return(96) ;
if (c==''ê'') return(95) ;
etc.

return(asc(c)-32);
```

À vous d'imaginer la procédure decode...

Codage

```
code:= (c)->{
if (c==''é'') return(100) ;
if (c==''è'') return(99) ;
if (c==''à'') return(98) ;
if (c==''ç'') return(97) ;
if (c==''ù'') return(96) ;
if (c==''ê'') return(95) ;
etc.
```

```
return(asc(c)-32);
```

À vous d'imaginer la procédure decode...

Ze programme

Ce qui donne au cœur du programme :

```
messcode:=concat(messcode,decode(irem(cle+code(message[j]),106)));
```

À vous de reconstituer le reste...

Ze programme

Ce qui donne au cœur du programme :

```
messcode:=concat(messcode,decode(irem(cle+code(message[j]),106)));
```

À vous de reconstituer le reste...