

Le décryptage du message trouvé chez « un autre monde »  
Code Vigenère

« DEEMX EDDAU IYFKW LTKEE TGNTK ECGMR GIMGD WSHIY IWJ »

« code=sapiens »

Ce message est crypté avec la méthode de Vigenère dont la clé est sapiens.

1 le cryptage

On ne considère que les caractères de *a* à *z* en minuscules. Les codes ASCII vont de 97 à 122. Pour chaque lettre du message crypté, on extrait le code ASCII auquel on retranche la clé. Seulement, si le code ASCII est inférieur strictement à 97, il faut repartir de 122.

Méthode 1 : Avec le tableur.

Voir le décryptage ci-après, il suffit d'adapter.

Méthode 2 : Algobox

Avec algobox, on utilise les fonctions length, substr, charCodeAt et String.fromCharCode

```
1 VARIABLES
2 message EST_DU_TYPE CHAINE
3 lettre EST_DU_TYPE CHAINE
4 long_message EST_DU_TYPE NOMBRE
5 long_cle EST_DU_TYPE NOMBRE
6 cle EST_DU_TYPE CHAINE
7 k EST_DU_TYPE NOMBRE
8
9 nb EST_DU_TYPE NOMBRE
10 compteur EST_DU_TYPE NOMBRE
11 nbc EST_DU_TYPE NOMBRE
12 decalage EST_DU_TYPE NOMBRE
13 compteur_cle EST_DU_TYPE NOMBRE
14 DEBUT_ALGORITHME
15 LIRE message
16 LIRE cle
17 long_cle PREND_LA_VALEUR cle.length
18 long_message PREND_LA_VALEUR message.length
19 compteur PREND_LA_VALEUR 1
20 compteur_cle PREND_LA_VALEUR 0
21 TANT_QUE (compteur<=long_message) FAIRE
22 DEBUT_TANT_QUE
23 nb PREND_LA_VALEUR message.charCodeAt(compteur-1)
24 SI (compteur_cle>=long_cle) ALORS
25 DEBUT_SI
26 compteur_cle PREND_LA_VALEUR 1
27 FIN_SI
```

```

28 SINON
29 DEBUT_SINON
30 compteur_cle PREND_LA_VALEUR compteur_cle+1
31 FIN_SINON
32 nbc PREND_LA_VALEUR cle.charCodeAtAt(compteur_cle-1)
33 decalage PREND_LA_VALEUR nbc-97
34 nb PREND_LA_VALEUR nb+decalage
35 SI (nb>122) ALORS
36 DEBUT_SI
37 nb PREND_LA_VALEUR nb-26
38 FIN_SI
39 lettre PREND_LA_VALEUR String.fromCharCode(nb)
40 AFFICHER lettre
41 compteur PREND_LA_VALEUR compteur+1
42 FIN_TANT_QUE
43
44
45
46 FIN_ALGORITHME

```

## 2 – le décryptage

Méthode 1 : Avec le tableur.

Voir le fichier `decodage_vigenere_tableur.ods` (ci-dessous extrait de grille d'après élèves))

lettre	décalage	code ascii								
a	0	97	message initial	d	e	e	m	x	e	d
b	1	98	clé	s	a	p	i	e	n	s
c	2	99	décalage	18	0	15	8	4	13	18
d	3	100	code ascii du message initial	100	101	101	109	120	101	100
e	4	101	code ascii-décalage	82	101	86	101	116	88	82
f	5	102	nouveau code ascii	108	101	112	101	116	114	108
g	6	103	lettre décodée	l	e	p	e	t	r	l
h	7	104								

On utilise ici les fonctions « `cod()` » et « `car()` » pour passer de lettre à ascii et inversement.  
 Voir le fichier `message_vigenere_codage.alg`

Méthode 2 : avec algobox

Voir le fichier `message_vigenere_decodage.alg`

```

1 VARIABLES
2 message EST_DU_TYPE CHAINE
3 lettre EST_DU_TYPE CHAINE
4 long_message EST_DU_TYPE NOMBRE
5 long_cle EST_DU_TYPE NOMBRE
6 cle EST_DU_TYPE CHAINE

```

```

7 k EST_DU_TYPE NOMBRE
8
9 nb EST_DU_TYPE NOMBRE
10 compteur EST_DU_TYPE NOMBRE
11 nbc EST_DU_TYPE NOMBRE
12 decalage EST_DU_TYPE NOMBRE
13 compteur_cle EST_DU_TYPE NOMBRE
14 DEBUT_ALGORITHME
15 LIRE message
16 LIRE cle
17 long_cle PREND_LA_VALEUR cle.length
18 long_message PREND_LA_VALEUR message.length
19 compteur PREND_LA_VALEUR 1
20 compteur_cle PREND_LA_VALEUR 0
21 TANT_QUE (compteur<=long_message) FAIRE
22 DEBUT_TANT_QUE
23 nb PREND_LA_VALEUR message.charCodeAt(compteur-1)
24 SI (compteur_cle>=long_cle) ALORS
25 DEBUT_SI
26 compteur_cle PREND_LA_VALEUR 1
27 FIN_SI
28 SINON
29 DEBUT_SINON
30 compteur_cle PREND_LA_VALEUR compteur_cle+1
31 FIN_SINON
32 nbc PREND_LA_VALEUR cle.charCodeAt(compteur_cle-1)
33 decalage PREND_LA_VALEUR nbc-97
34 nb PREND_LA_VALEUR nb-decalage
35 SI (nb<97) ALORS
36 DEBUT_SI
37 nb PREND_LA_VALEUR nb+26
38 FIN_SI
39 lettre PREND_LA_VALEUR String.fromCharCode(nb)
40 AFFICHER lettre
41 compteur PREND_LA_VALEUR compteur+1
42 FIN_TANT_QUE
46 FIN_ALGORITHME

```

### 3 Remarque

Il est intéressant de mettre en relation des groupes travaillant sur le Code de César d'une part et sur le code Vigenère d'autres parts et de leur faire comparer les fréquences des lettres que l'on obtient sur un même message. Le code de Vigenère permet d'obtenir un diagramme en bâtons beaucoup plus lissé.