Cryptologie

début mars

Évaluation

Ce TP est évalué. Cela ne change pas grand chose pour vous, c'est comme pour les TPs précédents, sauf que vous aurez une note à la fin (et que les corrigés ne sont pas en ligne pendant le TP). Vous serez évalués suivant les critères ci-dessous :

1. **avancement** : plus vous irez loin, meilleure sera votre note ;
2. **connaissances** : si vous me redemandez la syntaxe de la boucle `for`, ce n'est pas bon pour vous ;
3. **débrouillardise** : il est normal de ne pas tout connaître, par contre il est moins normal de ne pas savoir utiliser vos supports de cours, l'aide en ligne, les messages d'erreurs . . . ;
4. **esprit positif** : pas de "pfff de toute façon je comprends rien, je suis nul(le)".

Gardez tout ce que vous faites dans **un** fichier. Ça nous aidera pour vous évaluer.

Menu du jour

Aujourd'hui nous parlerons de codes secrets, de chiffrement, de décodage . . . thèmes très à la mode depuis (au choix)

- que notre gouvernement veut instaurer un filtrage de l'Internet
- la sortie du livre DA VINCI CODE

Ne vous inquiétez pas, on va commencer par les techniques utilisées il y a 2000 ans !

1 Le codage qui tue

Pour faire du codage, on préfère souvent travailler sur des nombres, et pas sur des lettres. Pour tout le TP, on va donc passer de la représentation en lettres à la représentation en nombres. Le code utilisé ne casse pas trois pattes à un lapin :

a	b	c	d	e	f	g	h	i	j	k	l	m
1	2	3	4	5	6	7	8	9	10	11	12	13
n	o	p	q	r	s	t	u	v	w	x	y	z
14	15	16	17	18	19	20	21	22	23	24	25	26

Exemple 1: $AMOUR = 1.13.15.21.18$

1. Créer un tableau `conversion` qui contienne les 26 lettres de l'alphabet romain ('A' en première case, 'B' en deuxième case ...);
2. Comment savoir, en utilisant ce tableau, la lettre codée par 15?
3. Compléter la fonction suivante qui décodera un message `msg`. `msg` est un tableau de nombres. La fonction doit afficher à l'écran le message décodé.

```
function decrypte(msg)
// for ...
```

```
endfunction
```

Aide: Vous pouvez la tester avec le mot 1.13.15.21.18 du dessus, c'est à dire [1 13 15 21 18] en **SCILAB**.

2 Le chiffre de César

Vous connaissez tous ce grand homme, auquel seul un petit village a résisté (encore et toujours). Ce que vous ne savez peut-être pas, c'est qu'il est l'auteur d'une technique de codage (pas très performante)! L'idée de base est que les deux individus qui veulent s'échanger un message connaissent tout deux une **clé** de chiffage, qui est en fait un nombre entre 1 et 26. Pour coder le message, on décale (vers la droite) toutes les lettres de ce nombre. Pour le décoder, on décale d'autant vers la gauche!

Exemple 2: Par exemple pour une clé de 2, 'A' valant 1, en le décalant de 2 vers la droite, on obtient 3 c'est à dire 'C'.

$AMOUR = 1.13.15.21.18 \Rightarrow 3.15.17.23.20 = COQWT$

Aide: Le code est cyclique, donc pour coder 'Y' valant 25 avec la clé 4, on obtient $25+4 - 26 = 3$ donc 'C'.

1. Compléter la fonction suivante qui applique le codage de César avec la clé 13 (c'est la clé qu'utilisait l'ami JULES). Dans ce cas encore, `msg` est un tableau de nombres. Le message codé sera stocké dans le tableau `code` sous forme d'entiers.

```
function code=cesar_crypte(msg)
// for ...
```

```
endfunction
```

2. Et maintenant, la fonction de décodage, toujours pour une clé de 13, qui n'est guère plus compliquée. Le message décodé sera stocké dans le tableau `decode`.

```
function decode=cesar_decrypte(msg)
// for ...
```

```
endfunction
```

3. vérifiez l'égalité $cesar_decrypte(cesar_crypte(msg)) == msg$.

3 Pourquoi César n'a pas conquit le monde

En fait le “chiffre de CÉSAR” n'est pas difficile à casser, pour deux raisons :

1. Il n'y a que 26 possibilités de clés, il suffit de toutes les tester ;
2. Une analyse de fréquences des lettres permet facilement de retrouver la valeur de la clé.

C'est l'analyse des fréquences qui va nous intéresser. On sait que en français, les lettres les plus fréquentes sont (dans l'ordre) le *e* suivi du *s*, du *a*, *i*, *t* Donc si un nombre (par exemple 8) apparaît souvent dans votre message codé, il y a de grandes chances que ce soit un 'E' (i.e. 5) et que la clé soit $8 - 5 = 3$.

1. Notre phrase de test sera 10.17.17.10.10.24.25.25.23.10.24.7.10.17.17.10, créer un tableau d'entier qui contienne cette phrase ;
2. afficher cette phrase codée avec la fonction `decrypte` du premier exercice. Pas terrible, non ?
3. Écrire une boucle `for` ou `while` qui compte combien de fois chaque nombre apparaît dans la phrase. Pour cela vous aurez besoin d'un tableau qui contient 26 zéros au départ, et que vous remplirez au fur et à mesure.
Aide: la fonction `zeros` pourra vous être utile.
4. Quels sont les nombres qui apparaissent le plus fréquemment ?
Aide: C'est une recherche de maximum . . .
5. En déduire la clé utilisée pour chiffrer le message du dessus. Que veut-il dire ?

4 Le chiffre du Che

Petit saut dans le temps, et nous voilà en pleine révolution communiste avec le camarade ERNESTO RAFAEL GUEVARA DE LA SERNA, alias “CHE GUEVARA”. Lui aussi utilisait des messages codés pour discuter avec . . . FIDEL CASTRO !

1. Tout comme dans notre exemple du début, il commençait par coder les lettres en nombres, en utilisant la table suivante :

a	b	c	d	e	f	g	h	i	j	k	l	m
6	38	32	4	8	30	36	34	39	31	78	72	70
n	o	p	q	r	s	t	u	v	w	x	y	z
76	9	79	71	58	2	0	52	50	56	54	1	59

Cette table vous est donnée pour information.

2. Il va maintenant falloir regrouper les nombres par paquet de 5 chiffres :
Exemple 3: `BONJOURPO = 38.9.76.31.9.52.58.79.9` $\Rightarrow$ `38976.31952.58799`

- (a) On va commencer par les regrouper par paquets de 1 chiffres : Si on a un message `msg` dans un tableau qui contient donc des nombres à 1 ou 2 chiffres, on fabrique un nouveau message avec uniquement des chiffres

Exemple 4: $BONJOURPO = 38.9.76.31.9.52.58.79.9 \Rightarrow 3.8.9.7.6.3.1.9.5.2.5.8.7.9.9$

- (b) avec ce nouveau tableau, on fabrique un tableau en regroupant les chiffres par paquets de 5 pour avoir un nouveau tableau

Exemple 5: $BONJOURPO = 38.9.76.31.9.52.58 \Rightarrow 3.8.9.7.6.3.1.9.5.2.5.8.7.9.9 \Rightarrow 38976.31952.58799$

3. Maintenant il faut choisir une clé, par exemple 25638. On va ajouter cette clé à chaque paquet de 5 du tableau du dessus, en jetant la retenue si la somme dépasse 100000.

Exemple 6: $BONJOURPO = 38.9.76.31.9.52.58 \Rightarrow 3.8.9.7.6.3.1.9.5.2.5.8.7.9.9 \Rightarrow 38976.31952.58799 \Rightarrow 53504.25638.84437$

Écrire la boucle `for` qui va faire cette transformation.

Aide: Pour décoder le message, il suffit de tout faire à l'inverse!

Pour votre culture, sur le corps du "Che" les boliviens ont trouvé un papier qui contenait les calculs intermédiaires nécessaires au codage!