Code de César

1. Le code de César

César, pour ses communications importantes à son armée, cryptait ses messages.

Le code de César est la méthode de cryptographie la plus ancienne communément admise par l'histoire. Il consiste en une **substitution mono-alphabétique :** chaque lettre est remplacée ("substitution") par une *seule* autre ("mono-alphabétique"), selon un certain décalage (appelé clé en cryptologie) dans l'alphabet.

1.1. Chiffrement

Ce que l'on appelle le chiffrement de César est un décalage des lettres.

Par exemple, pour chiffrer le mot NUMERIQUE avec un décalage de 3, le **N** devient **Q**, le **U** devient **X** pour ainsi obtenir le mot QXPHULTXH.

CLAIR	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
-> décalage = 3																										
CODE	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

1.2. Déchiffrement

Pour déchiffrer un code César, il suffit de décaler les lettres dans l'autre sens avec le même décalage qui a servi au chiffrement.

Par exemple, pour déchiffrer le mot LQIRUPDWLTXH avec un décalage de 3, le L devient I, le Q devient N pour ainsi former le mot INFORMATIQUE.

1.3. Sécurité du code

Niveau sécurité, le chiffre de César n'est pas fiable du tout, et ce pour deux raisons :

- Il n'existe que 26 façons différentes de crypter un message : puisqu'on ne dispose que de 26 lettres, il n'y a que 26 décalages possibles. Dès lors, des **attaques exhaustives** (tester toutes les décalages un à un) ne demande que très peu de temps.
- Le chiffre de César est très vulnérable à l'analyse des fréquences des lettres.

1.4. Exemples

✍ Chiffrer le mot CESAR avec un décalage (clé) de 3.

✍ Déchiffrer le code MXOHV qui utilise un décalage de 3.

✍ Chiffrer le mot ROMAIN avec un décalage (clé) de 10.

✍ Déchiffrer le code WQKBEY qui utilise un décalage de 10.

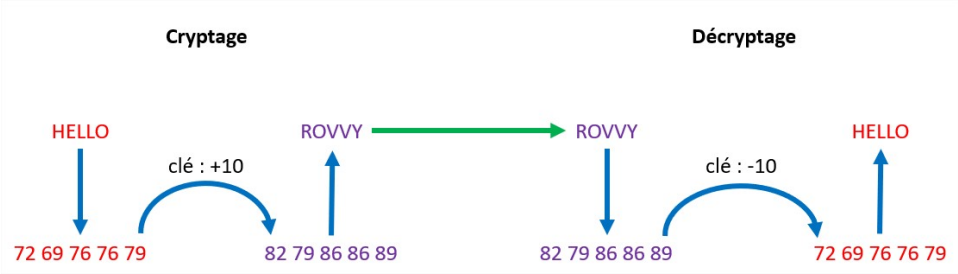
(cc) BY-NC-SA

2. Implémentation du code de César

2.1. Principe

Le principe utilisé va être le suivant :

- Pour crypter un message, chaque caractère du message à transmettre va être codé en ASCII, à ce code un décalage va être appliqué et ensuite le code obtenu va être retransformé en chaîne de caractère.
- Pour décrypter le message, la même méthode va être utilisée.



2.1. Conditions de l'implémentation

- Le chiffrement et le déchiffrement se fait sur les 26 caractères majuscules de l'alphabet sans accent.
- **Les espaces dans les chaînes de caractères sont conservés**.

2.2. Codage

✍ Réaliser une fonction `codage(text)` qui retourne la liste du code ASCII d'une chaîne de caractères.

2.1. Décodage

✍ Réaliser une fonction `decodage(code)` qui retourne la chaîne de caractères d'une liste de code ASCII.

2.2. Cryptage

✍ Réaliser une fonction de cryptage `crypt(text, dec)` qui retourne une chaîne de caractères codée à partir de la chaîne de caractères à coder et du décalage à effectuer.
Les fonctions `codage()` et `decodage()` seront appelées dans la fonction de cryptage.

2.3. Décryptage

✍ Réaliser une fonction de decryptage `crypt(text, dec)` qui retourne une chaîne de caractères décodée à partir de la chaîne de caractères codée et du décalage à effectuer.
Les fonctions `codage()` et `decodage()` seront appelées dans la fonction de décryptage.

2.4. Options

- Réaliser une interface qui demande l'opération à réaliser (cryptage ou décryptage) et le décalage à effectuer.
- Modifier les fonctions de cryptage et de décryptage afin de conserver la casse des caractères (minuscule ou majuscule).
- Améliorer la prise en compte de la chaîne de caractères en donnant la possibilité de saisir un texte avec accent (les caractères avec accent doivent être remplacés par des caractères sans accent. Par exemple, la chaîne de caractères été sera remplacée par ete. Lors du décryptage, les accents seront perdus).

Source : http://www.cryptage.org/chiffre-cesar.html

(cc) BY-NC-SA