

Mini-Rapport : Problème noté n°2

Explication de la seconde méthode de chiffrement/déchiffrement :

L'utilisateur renseigne dans un premier temps une clé de cryptage n .

Le cryptage se fait alors de la manière suivante :


- si le code ASCII de la lettre est pair alors on la crypte par la lettre située à $2n$ position après elle dans la table ASCII (de manière circulaire)
- si le code ASCII de la lettre est impair alors on la crypte par la lettre située à $2n$ position avant elle dans la table ASCII (de manière circulaire)

Le décryptage fonctionnera dans le sens inverse, c'est-à-dire :


- si le code ASCII de la lettre est pair alors on la crypte par la lettre située à $2n$ position avant elle dans la table ASCII (de manière circulaire)
- si le code ASCII de la lettre est impair alors on la crypte par la lettre située à $2n$ position après elle dans la table ASCII (de manière circulaire)

Ex :

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
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90



On décale de 6 positions ($2 \cdot n$) vers la droite si le code ASCII est pair



On décale de 6 positions ($2 \cdot n$) vers la gauche si le code ASCII est impair

Si on a $n=3$:

Alors si on crypte : A => U | B => H | Y => S | Z => F

Inversement si on décrypte : U => A | H => B | S => Y | F => Z

Explication du fonctionnement du menu :

void menu_cryptage (): Cette fonction demande à l'utilisateur de sélectionner la fonctionnalité qu'il désire à travers un menu. Le menu est relancé une fois qu'une fonctionnalité se termine (sauf quand l'utilisateur veut quitter). On a veillé, dans cette fonctionnalité et les suivantes, de redemander à l'utilisateur de saisir les données si elles sont erronées (saisie sécurisée).

void chiffrer_un_mot(): Cette fonctionnalité demande à l'utilisateur de saisir un mot à crypter puis une clé n à travers la fonction **saisir_mot()** et **saisir_cle()**. Elle crypte ensuite le mot à l'aide de **chiffrement(mot_saisi,n)**. La fonction **chiffrement** crypte caractère par caractère le mot rentré en paramètre grâce à la fonction **coder_lettre(mot[i], cle)**. Cette dernière fonction applique les règles de cryptage selon César en veillant bien qu'une minuscule reste une minuscule et une majuscule reste une majuscule. Enfin la fonctionnalité renvoie le mot crypté ainsi créé.

void dechiffrer_un_mot(): Cette fonctionnalité est la fonctionnalité inverse de **chiffrer_un_mot** et repose sur le même principe. Elle demande à l'utilisateur de saisir un mot à décrypter puis une clé n à travers la fonction **saisir_mot()** et **saisir_cle()**. On utilise ensuite de la même manière la fonction **dechiffrement** que **chiffrement** et **decoder_lettre** que **coder_lettre**. Enfin la fonctionnalité renvoie le mot décrypté ainsi créé.

void chiffrer_un_texte(): Cette fonctionnalité demande à l'utilisateur de saisir le nom du fichier à crypter, le nom du futur fichier créé puis une clé n à travers la fonction **choix_texte()**, **saisir_fichier_nouveau()** et **saisir_cle()**. Ensuite elle ouvre le premier fichier en mode lecture et crée le second en mode écriture. Elle saisit en première ligne du nouveau fichier la clé de lecture n, puis inscrit l'autre fichier en crypté en procédant caractère par caractère grâce à la fonction **coder_lettre**. Elle renvoie alors si l'opération a bien pu être effectuée.

void dechiffrer_un_texte():() Cette fonctionnalité est la fonctionnalité inverse de **chiffrer_un_texte** et repose sur le même principe. Elle demande à l'utilisateur de saisir le nom du fichier à décrypter et le nom du futur fichier créé à travers la fonction **choix_texte()** et **saisir_fichier_nouveau()** (Elle ne demande pas la clé de lecture, cette dernière sera en première ligne du fichier crypté). Ensuite elle ouvre le premier fichier en mode lecture et crée le second en mode écriture. Elle lit en première ligne du fichier crypté la clé de lecture n, puis lance le décryptage du fichier crypté en procédant caractère par caractère grâce à la fonction **decoder_lettre**. Elle renvoie alors si l'opération a bien pu être effectuée.

void autre_methode(): Cette fonctionnalité donne un sous menu permettant d'utiliser l'autre méthode de cryptage dont le principe a été détaillé ci-dessus. Les différentes fonctionnalités qu'elle propose (**chiffrer_un_mot_alternatif()**, **dechiffrer_un_mot_alternatif()**, **chiffrer_un_texte_alternatif()** et **dechiffrer_un_texte_alternatif()**) sont identiques au menu classique, le codage et le décodage est juste réadapté pour cette nouvelle méthode de cryptage.