

## Projet Python : Le pays du chiffrement

Nous avons commencé notre projet dans l'idée de créer un programme permettant de chiffrer un texte en code César et de le décrypter sans avoir sa clé. Nous avons ensuite élargi notre projet en y rajoutant du morse, du braille, le code des rois de France et le code lapin.

Notre projet se présente donc comme une interface Tkinter. Dans le menu, l'utilisateur a plusieurs choix possibles :

- Dans la colonne « Code César », il peut chiffrer un texte avec une clé (cette clé doit être un nombre entier relatif, il n'y a pas de maximum ou de minimum). Il peut également rentrer un texte chiffré en code César dans la partie « décodage », sans avoir sa clé. Le programme rendra alors les textes déchiffrés les plus probables. En cas d'échec de ce programme, l'utilisateur peut toujours afficher les 26 différents décalages.

- Dans la colonne « Morse », l'utilisateur peut rentrer son texte qui sera immédiatement converti en morse, cela fonctionne également dans le sens contraire. Il peut également rentrer un texte en morse dans la partie « bip », le programme va ainsi produire les bips correspondants.

- Dans la colonne « Code des rois de France », l'utilisateur rentre son texte avec une clef (une chaîne de caractères). Le code des rois de France est un code César plus élaboré, ayant pour clef un ou plusieurs caractères. Les lettres du texte d'origine seront décalées selon une valeur attribuée aux lettres de la clef. Par exemple, si on veut coder « mathurine » avec la clef « abc », le m sera codé avec un décalage de 0 (correspondant au a), il ne sera donc pas transformé. Le a sera codé avec un décalage de 1 (correspondant au b) et deviendra donc un z ; le t sera décalé de 2 et deviendra un r. Puis la clef est reprise au début. Ainsi, mathurine deviendra mzhrtipmc. La partie « décodage » permet de décoder un texte chiffré avec le code des rois de France si et seulement si l'utilisateur connaît la clef (il est très difficile de déchiffrer ce code sans avoir la clef).

- Dans la colonne « Braille », les parties « codage » et « décryptage » fonctionnent comme pour le morse mais rendent le texte donné en braille.

- Dans la colonne « Code lapin », l'utilisateur peut chiffrer et décrypter un texte en utilisant le « code lapin ». Voici comment il fonctionne : dans une grille de 5 X 5 on écrit les lettres de l'alphabet (en omettant le W) et chaque lettre est codée par les coordonnées de la grille. Les coordonnées ne sont pas des chiffres mais le mot LAPIN.

	L	A	P	I	N
L	a	b	c	d	e
A	f	g	h	i	j
P	k	l	m	n	o
I	p	q	r	s	t
N	u	v	x	y	Z

- Dans la colonne « Un peu d'histoire », on donne quelques explications sur la création ou le fonctionnement des différentes méthodes de chiffrement proposées.

Nous avons eu de nombreux problèmes en écrivant notre code python, la plupart concernant la fonction « pourcentages », permettant de décrypter un texte chiffré en code César sans sa clé. Après une première fonction marchant dans la majorité des cas, nous avons pris une nouvelle approche donnant plus d'options. Cette dernière fonction est plus efficace que la première.

Nous avons également eu un petit problème par rapport au braille. En effet, le signe braille de certains chiffres correspondent à celui de lettres accentuées. Nous n'avons donc pas transformé les lettres accentuées directement en braille, on les transforme d'abord dans leur équivalent non-accentué.

Le programme est très long, ce qui pose des difficultés pour trouver les endroits où il y a des erreurs.

Pour l'interface graphique, il était souvent difficile de trouver une solution aux problèmes rencontrés, et il fallait chercher longtemps le bon paramètre à changer pour atteindre nos objectifs.

Nous avons également cherché longtemps la fonction permettant de produire un bip.