LES TROPHÉES NSI

Édition 2024

DOSSIER DE CANDIDATURE

PRÉSENTATION DU PROJET

Le chiffre clé

nom de votre projet :	Le chiffre clé
membres de l'équipe :	Simon GALLAND
membres de l'équipe :	Maxime ANTOINE
niveau d'étude :	Première
établissement scolaire :	Lycée Notre dame – 21 000 Dijon
enseignante/enseignant de NSI :	M. P.MOREAU

> PRÉSENTATION GÉNÉRALE :

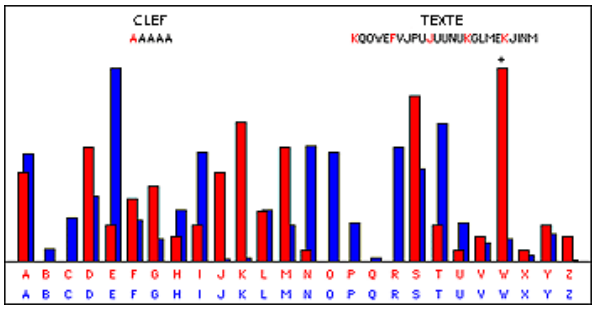
Le projet « Le chiffre clé » se concentre sur le décryptage automatique d'un message crypté par la méthode de Vigenère, mais **sans connaître la clef**. Cette méthode prend en compte les 26 lettres de l'alphabet latin, sans caractère spécial, et repose sur une clef (= une chaîne de caractère).

Le cryptage Vigenère consiste à un chiffrement d'un texte dit "clair" : normal, que nous trouvons couramment. Chaque caractère de la clef est interprété en décalage selon la position dans l'alphabet (A=0, B=1, C=2, etc.), les décalages étant appliqués successivement et cycliquement.

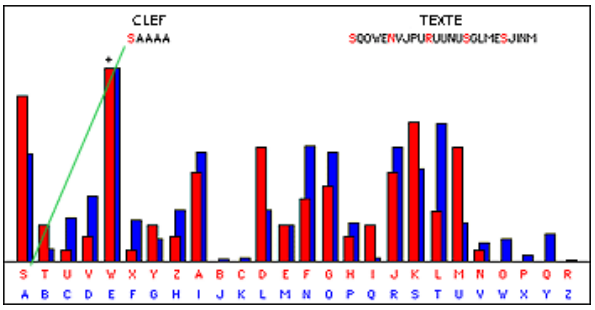
Clair	C	H	I	F	F	R	E	D	E	V	I	G	E	N	E	R	E
Clef	B	A	C	H	E	L	I	E	R	B	A	C	H	E	L	I	E
Décalage	1	0	2	7	4	11	8	4	17	1	0	2	7	4	11	8	4
Chiffré	D	H	K	M	J	C	M	H	V	W	I	I	L	R	P	Z	I

Nous pouvons remarquer que la clef est répétée en dessous du texte clair. Le but est d'obtenir le texte chiffré (=crypté).

Les trois points importants de notre projet sont, premièrement de trouver la longueur de la clef, via le test de Friedman. Deuxièmement, la taille de la clef permet de trouver la clef ; la lettre qui apparaît le plus de fois dans un décalage est rapprochée à la lettre E.



Situation initiale



Après le décalage qui minimise les écarts entre les bâtons des histogrammes bleu (théorique) et rouge.

À gauche, nous pouvons observer que la récurrence de la lettre W se rapproche de celle de E. Ainsi, nous effectuons un décalage afin que le E et W soient à la même place. Enfin, nous récupérons la première qui constitue la première du mot-clef (ici la lettre S). Troisièmement, nous décryptons le message chiffré avec la clef que l'on a obtenue précédemment.

Nous avions déjà codé les codes César et Vigenère plus tôt dans l'année. Nous nous étions demandé s'il était possible de "craquer" le code Vigenère sans connaître la clef. Pour cela nous avons consulté une documentation très précise afin d'en apprendre plus sur le test de Friedman et autres étapes. De plus, nous avons totalement été à l'initiative de toutes nos fonctions que l'on a créées.

> ORGANISATION DU TRAVAIL :

Dans le cadre de notre collaboration, nous avons réparti les tâches de manière équilibrée et complémentaire. Simon GALLAND et Maxime ANTOINE se sont partagé la responsabilité en s'occupant de 5 fonctions chacun, ce qui a favorisé une dynamique de travail collaborative. Notre approche était centrée sur la collaboration et l'échange constant d'idées et de réflexions, ce qui nous a permis de progresser ensemble tout au long du projet.

Pour mener à bien notre projet, nous avons opté pour l'utilisation de plusieurs outils en ligne. Nous avons ainsi utilisé la plateforme "REPLIT" pour coder simultanément et Google Docs pour échanger toutes les informations et documents nécessaires comme par exemple la présentation du projet ou encore le dossier technique. Cette combinaison d'outils nous a offert un environnement de travail efficace et collaboratif, nous permettant de coordonner nos efforts et de partager nos avancées en temps réel.

Dans le cadre de nos rôles respectifs, Simon s'est chargé de la gestion du dossier technique. Grâce à ses capacités en matière de gestion informatique et de manipulation de fichiers, il a su assurer une organisation optimale de nos ressources et de nos données. De son côté, Maxime a pris en charge la rédaction et la description détaillée du projet. Sa préférence pour l'explication et la rédaction lui a permis de mettre en mots clairs et précis les objectifs, les fonctionnalités et les implications de notre projet.

La documentation technique a joué un rôle crucial dans notre processus de développement. Pour approfondir certains aspects et fonctions spécifiques, nous avons également utilisé l'outil PyCharm, qui nous a offert un environnement de développement intégré complet et adapté à nos besoins pour pouvoir trouver le problème de notre code plus simplement .

Malgré les contraintes de temps, nous avons consacré six semaines à ce projet, en travaillant environ six heures par semaine en dehors des heures de cours. Cette organisation nous a permis de progresser de manière régulière et efficace, tout en respectant les délais fixés et en assurant la qualité de notre travail.

> LES ÉTAPES DU PROJET :

Nous étions conscients des étapes suivantes (ci-dessous) et nous avons procédé fonction par fonction, au fur et à mesure.

Étape 1 : ÉTUDE DU TEST DE FRIEDMAN – TAILLE DE LA CLEF

Le test de Friedman repose sur la métrique de l'indice de Coïncidence (inventé en 1920). L'indice de Coïncidence (IC) est la probabilité que deux lettres choisies aléatoirement dans un texte soient identiques. Pour calculer cet indice, soient :

n = nombre de lettres du texte ;

n1 = nombre de A dans le texte ;

n2 = nombre de B dans le texte ;

n3 = nombre de C dans le texte ; … ;

n26 = nombre de Z dans le texte.

$$IC = \sum_{i=1}^{26} \frac{n_i (n_i - 1)}{n (n - 1)}$$

Ainsi, pour calculer la probabilité de tirer deux lettres identiques, il faut faire la somme des 26 possibilités (programme de 1ère) :

Dans un premier temps, nous découpons le message crypté selon des intervalles (de 1 en 1, de 2 en 2, …). Par exemple, pour le message crypté "JIEOZCN", l'intervalle de 2 est : "JEZN". Dans un deuxième temps, nous calculons l'IC pour chaque décalage, et nous analysons s'il se rapproche de l'IC de la langue français (établi à 0,074). Dans un troisième temps, la longueur de la clé correspond au décalage qui est proche de l'IC français.

Étape 2 : TROUVER LA CLEF

Nous récupérons ensuite les lettres qui composent les décalages, c'est-à-dire les messages cryptés découpés, ainsi que leur occurrence (le nombre de fois qu'une lettre apparaît). Puis, pour chaque décalage, on récupère la lettre qui apparaît le plus souvent. Nous comparons ensuite sa place dans l'alphabet par rapport à la lettre E (E = 4, car A = 0, B = 1). Donc la différence entre la place de la lettre (qui apparaît le plus souvent) et la lettre E nous donne l'emplacement de la lettre de la clef. Voir schéma n°2 page 2 . Puis nous refaisons cette méthode pour chaque "décalage", et nous trouvons à la fin la clef.

Étape 3 : DÉCRYPTER

Maintenant que nous avons identifié la clé et sa longueur, nous pouvons procéder au processus de décryptage du message crypté. Ce processus implique une manipulation des caractères du message à l'aide de la clé. Pour illustrer ce processus, prenons un exemple simple à partir de notre première observation.

Supposons que nous ayons identifié que la clé est "B" et que la première lettre du message crypté est "D". Maintenant, pour déchiffrer cette première lettre, nous effectuons une opération de décalage en soustrayant l'index de la première lettre du message crypté ("D" est d'index 3) de l'index de la première lettre de la clé ("B" est d'index 1). Pour clarifier cela, référons-nous au schéma suivant :

Clair	C
Clef	B
Décalage	1
Chiffré	D

- L'index de la lettre "D" est 3.

- L'index de la lettre "B" est 1.

Donc, pour déchiffrer la lettre "D", nous effectuons l'opération suivante : 3 - 1 = 2

Grâce à ce calcul, nous pouvons enfin retrouver le premier caractère du message décrypté . En conséquence, nous appliquons un décalage de 1 à la lettre "C", ce qui nous donne la lettre déchiffrée. Ce processus est répété pour chaque lettre du message, en appliquant le décalage approprié déterminé par la clé.

> FONCTIONNEMENT ET OPÉRATIONNALITÉ

Le projet est bien complet, et les trois étapes précédentes sont fonctionnelles. Nous n'avions pas prévu au départ une interface. Nous avons utilisé Pycharm pour se focaliser sur une fonction par exemple, quand nous avons préparé le dossier, et éviter des lignes. Nous avons réfléchi à ce qui pouvait causer une erreur dans le programme, les éventuels non-respect des pré-requis, etc…

Cependant, nous retirons les caractères du message crypté, notamment les espaces. Par conséquent, le message décrypté n'aura pas d'espace et les mots le composant seront tous à la suite. Une amélioration serait de prendre en compte ces espaces, mais cela prendrait énormément de temps. Enfin, sachant que les "ê", "é", "è", les "à", "â", etc…ne composent l'alphabet français, ils ne sont pas compris dans l'alphabet de notre programme et ainsi, ils sont directement supprimés, ce qui pourrait nuire à la bonne exécution de notre programme.

Enfin, il existe de nombreux qui comprennent peu ou pas de caractère 'E'. Par exemple le livre 'La Disparition de Georges Perec".

En somme, nous avons réalisé nos attentes principales.

> OUVERTURE :

Nous pensons qu'avec un peu plus de temps, nous pouvons optimiser le programme au maximum. De plus, nous envisageons de créer une interface plutôt simple, avec un lien pour crypter n'importe quel message, ainsi qu'une zone de texte pour entrer le message. Il pourrait également avoir une interface pour aide à l'utilisation.

Nous nous sommes focalisés sur la technique du test de Friedman et son indice de coïncidence. Cependant, il en existe d'autres. Pour trouver la taille de la clé, Charles Babbage a développé une méthode qui consiste à retrouver des structures du messages qui se ressemblent :

KQOWE FVJPU JUUNU KGLME KJINM WUXFQ MKJBG WRLFN FGHUD **WUU**MB SVLPS NCMUE KQCTE SWR**EE K**OYSS IWCTU AXYOT APXPL WPNTC GOJBG FQHTD **WXIZA YG**FFN SXCSE YNCTS SPNTU JNYTG GWZGR **WUU**NE JUUQE APYME KQHUI DUXFP GUYTS MTFFS H**NUOC ZGM**RU WEYTR GKMEE DCTVR ECFBD JQCUS WVBPN LGOYL SKMTE FVJJT WWMFM WPNME MTMHR SPXFS SKFFS T**NUOC ZGMDO EOYEE K**CPJR GPMUR SKHFR SEIUE VGOYC **WXIZA YG**OSA ANY**DO EOY**JL WUNHA MEBFE LXYVL WNOJN SIOFR WUCCE SWKVI D**GMU**C GOCRU WGNMA AFFVN SIUDE KQHCE UCPFC MPVSU DGAVE MNYMA MVLFM AOYFN TQCUA FVFJN XKLNE IWCWO DCCUL WRIFT W**GMU**S WOVMA TNYBU HTCOC WFYTN MGYTQ MKBBN LGFBT WOJFT WGNTE JKNEE DCLDH WTVBU VGFBI JG

Nous retrouvons l'espacement (en caractère) entre chaque même structure. Par exemple, il y a 95 caractères entre les deux structures 'WUU'. Les facteurs premiers du nombre de caractères entre deux débuts de séquences figurent dans le tableau. Il apparaît dans le tableau que toutes les distances sont divisibles par 5. Tout se cale parfaitement sur un mot-clef de 5 lettres.

		Longueurs de clefs possibles								
Séquence répétée	Distance	2	3	4	5	9	10	15	19	20
WUU	95				x				x	
EEK	200	x		x	x		x			x
WXIZAYG	190	x			x		x		x	
NUOCZGM	80	x		x	x		x			x
DOEOY	45		x		x	x		x		
GMU	90	x	x		x	x	x	x		

Nous avons perfectionné nos compétences en dictionnaire. Ensuite nous avons modélisé grâce aux Mathématiques (symbole somme). En outre, nous connaissons au mieux la méthode de Vigenère. Enfin, nous avons appris à réaliser l'architecture du programme (dossier technique), fichier csv, le 'README', etc…