

SUPINFO Academic Dept.

**Arithmétique et cryptographie**

Mini - Projet

Programmation de quelques algorithmes cryptographiques

Version 1.0

Last update: 18/02/2016

Use: Students/Staff

Author: Laurent GODEFROY

SOMMAIRE

0 Préambule 3

1 Le chiffre du livre 4

1.1 Généralités sur ce chiffre 4

1.2 Implémentation en Python 5

2 Chiffre de Playfair 6

2.1 Généralités sur ce chiffre 6

2.2 Implémentation en Python 7

3 Chiffre de Collon 9

3.1 Généralités sur ce chiffre 9

3.2 Implémentation en Python 10

# Préambule

Cet examen est à réaliser par groupes de deux étudiants. Dans l’unique cas où le nombre d’étudiants de la promotion est impair, un et un seul groupe de trois est autorisé.

Toute forme de plagiat ou utilisation de codes disponibles sur internet ou tout autre support, même de manière partielle, est strictement interdite et se verra sanctionnée d’un 0, d’une mention « cheater », et le cas échéant d’un conseil de discipline.

Vous devrez envoyer votre projet par mail à **votre formateur** avant le **dimanche 20 mars 2016** à 23h59, heure locale. Au delà de cette date et heure votre note sera de 0. Vous comprimerez vos codes sources dans une archive au format « .zip » que vous nommerez en respectant la convention **ID-1ARI-votreCampus-MP** (exemple : 66280-1ARI-Tours-MP). L’objet du mail sera ce même nom. Vous mettrez également **en copie** la boîte [1ARI@supinfo.com](mailto:1ARI@supinfo.com). Si vous ne mettez pas cette dernière adresse en copie aucune réclamation ultérieure ne sera recevable.

Ce mini-projet donnera lieu à des soutenances qui se dérouleront la semaine du 21 mars 2016. Vos horaires de passages vous seront communiqués par votre campus.

Les soutenances sont également par groupes de deux. Elles dureront **20 minutes** pendant lesquelles vous montrerez à votre examinateur le bon fonctionnement de votre programme en en faisant la démonstration. Si vous n’avez pas implémenté le projet dans sa totalité, vous exposerez les parties fonctionnelles.

Pour appuyer votre présentation, vous devrez préparer un fichier de type Powerpoint, dans lesquels vous expliquerez les points du code que vous jugez les plus importants et significatifs. Il n’est pas nécessaire d’envoyer ce fichier à votre examinateur, ce dernier le découvrira le jour de la soutenance. Une communication précisant tout cela vous sera envoyé courant mars.

D’autre part, **n'hésitez pas à utiliser le forum pour échanger sur ce projet, j'ai ouvert un fil de discussion à cet usage.**

On devra nécessairement utiliser la librairie graphique Pygame dans la partie 2. L’usage d’une autre librairie ne sera pas pris en compte.

On pourra si on le souhaite ajouter des fonctions et procédures à celles demandées.

**Remarque importante**: des sous-programmes écrits dans une partie pourront bien sûr être importés dans une autre.

# Le chiffre du livre

## Généralités sur ce chiffre

**Remarque importante**: aucun code n’est demandé dans ce paragraphe qui n’est qu’explicatif.

Le but de cette partie est d'implémenter le chiffre du livre, aussi appelé système du dictionnaire, et qui est un chiffre de substitution homophonique.

Pour procéder au chiffrage, il faut se donner une clé, qui est ici un texte assez long, historiquement une page fixée d’un livre.

On chiffre chacune des lettres du texte en clair en prenant au hasard un des indices de cette lettre dans la clé. L’indexation se fait en ne prenant pas en compte les espaces, lettres accentuées, signes de ponctuations, etc.

Si des lettres ne figurent pas dans la clé, on les chiffrera par les nombres suivants le dernier indice.

**Exemple** : on prend comme clé le texte de présentation du campus SUPINFO à Tours.

« L'établissement d'enseignement supérieur SUPINFO à Tours, situé en plein centre-ville, offre un espace de travail et de détente des plus appropriés. L'existence d'un Campus SUPINFO à Tours permet aux jeunes tourangeaux mais aussi aux jeunes d'Indre-et-Loire d'étudier l'informatique et le numérique à proximité de leur famille.

L’équipe administrative du Campus se met un point d'honneur à accompagner les étudiants SUPINFO tout au long de leur cursus et prend à cœur de les placer dans les meilleures conditions d’apprentissage pour conduire chacun d'entre eux à la réussite. »

On veut chiffrer « SUPINFO ». La lettre ‘S’ pourra être chiffrée par 6,7,16,26,34,45,…,449 ou 450. La lettre ‘U’ par 27,32,35,43,…,443 ou 448. Etc.

Ainsi un chiffrement possible de « SUPINFO » est 26 144 112 330 159 140 143.

A noter que les lettres ‘K’, ‘W’, ‘X’ et ‘Y’ ne figuraient pas dans le texte servant à constituer la clé. Elles auraient été chiffrées en 454, 455, 456 et 457, car le dernier indice du texte était 453.

Le déchiffrement est immédiat.

## Implémentation en Python

**Il vous est fortement recommandé de lire l’intégralité de cette partie avant de commencer à coder. Les travaux demandés sont mis en évidence avec une couleur bleue.**

**Dans un fichier que l’on nommera « chiffreDuLivre.py », implémenter les sous-programmes suivants :**

* Une fonction prenant en paramètre une chaîne de caractères, et qui la retourne après avoir supprimé les espaces, signes de ponctuations, lettres accentuées, etc.
* Une fonction prenant en paramètre une chaîne de caractères, et qui retourne un dictionnaire dont les clés seront les lettres de l’alphabet en majuscules. La valeur associée à une clé sera la liste des indices de cette clé dans la chaîne. Pour calculer les indices, on ne tiendra pas compte des espaces, signes de ponctuations, lettres accentuées, etc.
* Une fonction prenant en paramètre une lettre et une clé (i.e. un dictionnaire de la forme précédente), et qui retourne un nombre chiffrant la lettre selon l’algorithme du chiffre du livre.
* Une fonction prenant en paramètre un texte et une clé (i.e. un dictionnaire de la forme précédente), et qui retourne une liste de nombres chiffrant le texte selon l’algorithme du chiffre du livre.
* Des fonctions similaires aux deux précédentes mais permettant cette fois le déchiffrement.

**Utilisation de ces procédures et fonctions :**

* Déchiffrer le texte ‘chiffre1.txt’ avec la clé ‘cle1.txt’. On accèdera aux données de ces fichiers textes avec les fonctions usuelles de lecture, et non via un copier/coller dans le script.
* Chiffrer un texte de votre choix avec une clé de votre choix également et joignez la clé et le texte chiffré à votre projet sous la forme de fichiers “.txt“.

**Deux petites questions :**

* Que pensez-vous de la sécurité de cet algorithme ?
* Quels sont ses principales qualités et défauts ?

**Bonus :**

* Améliorer ce qui précède pour prendre en compte les espaces et les lettres accentuées sans avoir à les supprimer.

# Chiffre de Playfair

## Généralités sur ce chiffre

**Remarque importante**: aucun code n’est demandé dans ce paragraphe qui n’est qu’explicatif.

Le but de cette partie est d'implémenter le chiffre de Playfair, qui est un chiffre de substitution polygrammique. Rappelons ici son fonctionnement.

Pour procéder au chiffrage, il faut se donner une clé, qui est ici une répartition de 25 lettres de l’alphabet (toutes les lettres excepté le ‘w’) dans un carré de 5 cases sur 5. Par exemple :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| s | e | v | m | a |
| t | k | p | y | n |
| l | x | z | b | h |
| d | q | i | c | u |
| j | f | o | r | g |

Dans le texte à chiffrer on va tout d’abord remplacer tous les ‘w’ par des ‘v’, puis on va découper le texte en blocs de deux lettres. Si un des bigrammes obtenus est composé de deux lettres identiques, on va insérer entre elles un ‘x’. Une fois cela fait, si le nombre de lettres du texte est impair, on va ajouter à la fin un ‘x’ pour constituer le dernier bigramme.

**Par exemple**, la chaîne « hello world » va être transformée en « he lx lo vo rl dx ».

Voici maintenant les règles de chiffrement d’un bigramme :

* Si les deux lettres à chiffrer sont sur les sommets opposés d’un rectangle, les lettres chiffrées seront sur les deux autres sommets. Ainsi le bigramme « he » sera chiffré en « XA ». A noter que la première des deux lettres chiffrées est toujours sur la même ligne que la première des deux lettres en clair. Ainsi « eh » serait chiffré en « AX ».
* Si les deux lettres sont situées sur une même ligne on prend les deux lettres situées juste à leur droite. Si les deux lettres sont situées sur une même colonne, on prend les deux lettres situées juste en dessous d’elles. Par exemple « lx » sera chiffré en « XZ » et « vo » sera chiffré en « PV ».

**Suite de l’exemple**: la chaîne « hello world » transformée dans un premier temps en « he lx lo vo rl dx » va être chiffrée avec la clé précédente en « XA XZ ZJ PV JB QL ». Que l’on peut remettre en forme en utilisant les conventions usuelles : « XAXZZ JPVJB QL ».

Le déchiffrement s’obtient en appliquant ces règles « à l’envers ».

## Implémentation en Python

**Il vous est fortement recommandé de lire l’intégralité de cette partie avant de commencer à coder. Les travaux demandés sont mis en évidence avec une couleur bleue.**

**Dans un fichier que l’on nommera « chiffreDePlayfayr.py », implémenter les sous-programmes suivants :**

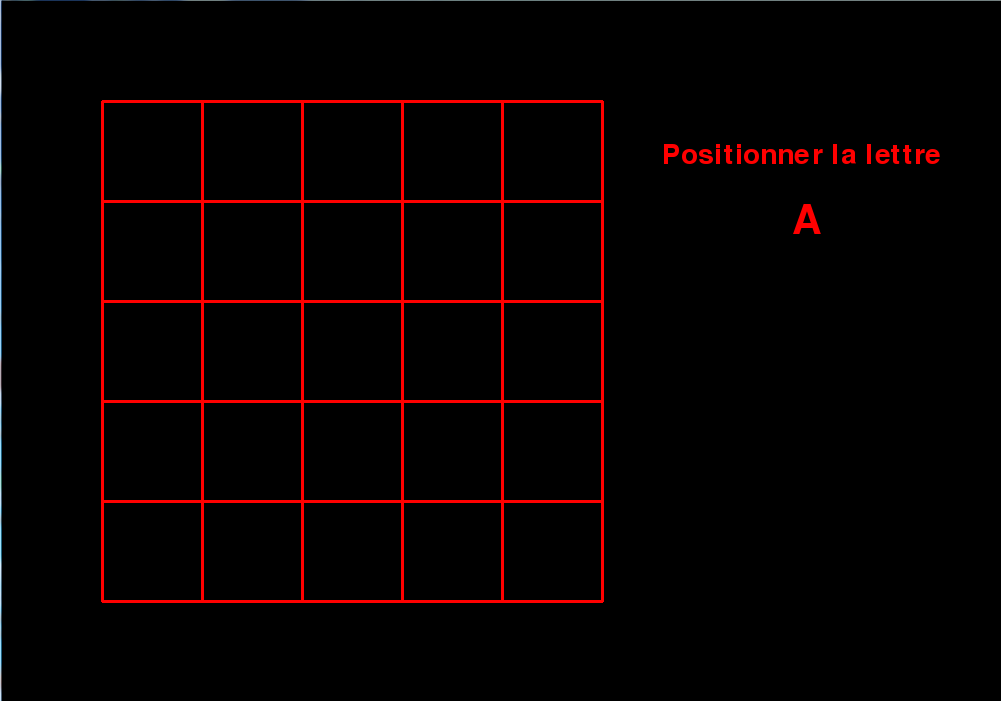
* Une fonction prenant en paramètre une chaîne de caractères, et qui la retourne après remplacé chacun des ‘w’ (s’il y en a) par un ‘v’.
* Une fonction prenant en paramètre une chaîne de caractères, et qui la retourne après avoir inséré à bon escient des ‘x’ afin d’éviter d’avoir des bigrammes de deux lettres identiques.
* Une fonction prenant en paramètre une chaîne de caractères, et qui la retourne après avoir ajouté un ‘x’ à la fin si le nombre de lettres est impair.
* Une fonction prenant en paramètre une chaîne constituée des 25 lettres en minuscules de l’alphabet privé du ‘w’ (chacune de ces 25 lettres n’apparaissant qu’une et une seule fois). Elle retourne une liste à deux dimensions de 5 lignes et 5 colonnes, remplie ligne par ligne à l’aide de la chaîne passée en paramètre. Par exemple la chaîne « sevmatkpynlxzbhdqicujforg » constituera le carré de la sous-partie précédente.
* Une fonction prenant en paramètres une liste à deux dimensions de 5 lignes et 5 colonnes, ainsi qu’un caractère que l’on supposera présent une et une seule fois dans la liste. Elle retourne les indices des lignes et colonnes du caractère dans la liste.
* Une fonction prenant en paramètres deux couples d’indices compris entre 0 et 4, et qui retourne deux couples d’indices entre 0 et 4, calculés selon les règles du chiffre de Playfair. Par exemple si [2,4],[0,1] sont passés en paramètre la fonction retournera [2,1],[0,4]. S’il s’agit de [0,2],[4,2] on obtiendra [1,2],[0,2].
* Une fonction prenant en paramètre un bigramme et une clé (i.e. une liste de la forme précédente), et qui retourne le bigramme chiffré selon l’algorithme de Playfayr.
* Une fonction prenant en paramètre un texte et une clé (i.e. une liste de la forme précédente), et qui retourne le texte chiffré selon l’algorithme de Playfayr. On le mettra en forme en respectant les conventions habituelles.
* Voir comment utiliser/modifier les fonctions précédentes pour procéder au déchiffrement d’un bigramme puis d’un texte.

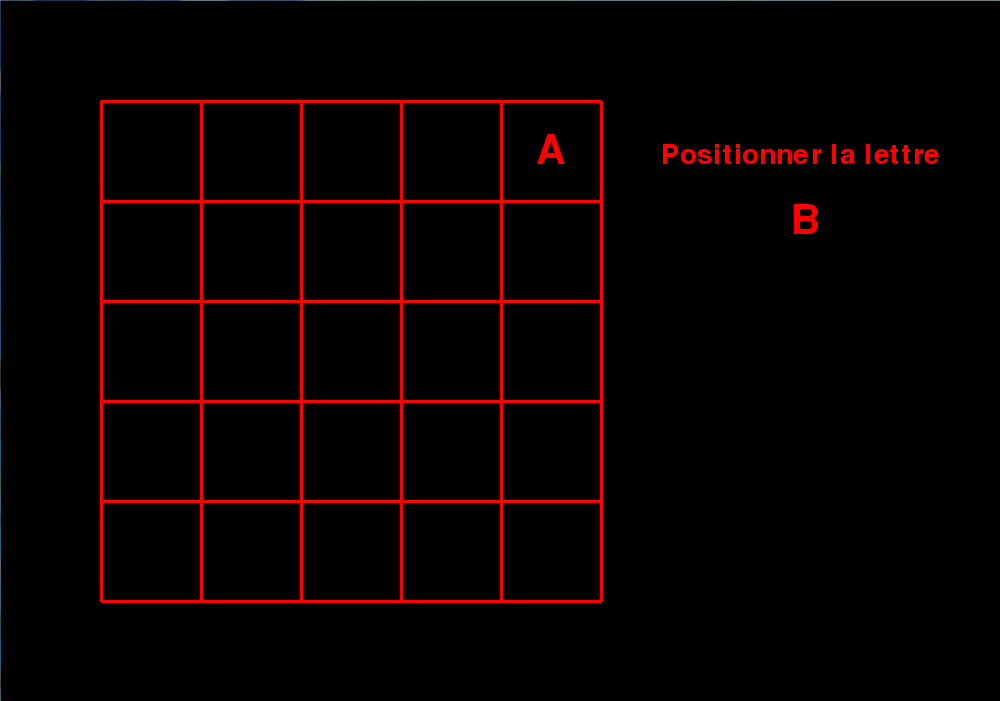
**Utilisation de ces procédures et fonctions :**

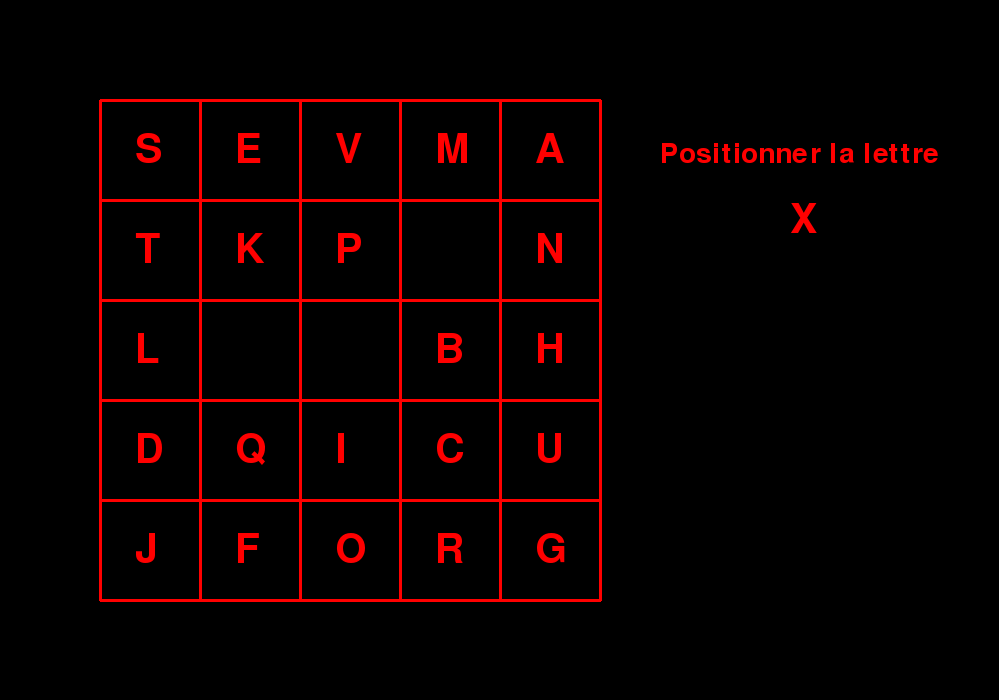
* Déchiffrer le texte ‘chiffre2.txt’ avec la clé ‘cle2.txt’. On accèdera aux données de ces fichiers textes avec les fonctions usuelles de lecture, et non via un copier/coller dans le script.
* Chiffrer un texte de votre choix avec une clé de votre choix également et joignez la clé et le texte chiffré à votre projet sous la forme de fichiers “.txt“.

**Complément (non facultatif) : une GUI pour saisir la clé**

* Implémenter une interface graphique permettant de saisir la clé lettre par lettre, en cliquant avec la souris sur la position souhaitée. Voici un exemple de rendu :







**Deux petites questions :**

* Que pensez-vous de la sécurité de cet algorithme ?
* Quels sont ses principales qualités et défauts ?

# Chiffre de Collon

## Généralités sur ce chiffre

**Remarque importante**: aucun code n’est demandé dans ce paragraphe qui n’est qu’explicatif.

Le but de cette partie est d'implémenter le chiffre de Collon, qui est un chiffre de substitution tomogrammique. Présentons ici son fonctionnement.

Pour procéder au chiffrage, il faut se donner une clé, qui est ici un entier, par exemple 7, et une répartition de 25 lettres de l’alphabet (toutes les lettres excepté le ‘w’) dans un carré de 5 cases sur 5.

**Par exemple**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| s | e | v | m | a |
| t | k | p | y | n |
| l | x | z | b | h |
| d | q | i | c | u |
| j | f | o | r | g |

Dans le texte à chiffrer on va tout d’abord remplacer tous les ‘w’ par des ‘v’.

On va ensuite chiffrer chacune des lettres du texte par un bigramme en procédant comme ceci : on commence par repérer le numéro i de la ligne de la lettre et le numéro j de la colonne de la lettre. On forme alors un bigramme avec le i-ème élément de la première colonne et le j-ème élément de la dernière ligne.

**Exemple** : ‘h’ est chiffré en « LG », ‘e’ est chiffré en « SF », ‘l’ est chiffré en « LJ ».

Pour chiffrer un texte, on chiffre chacune de ses lettres en bigrammes que l’on écrit sur 2 lignes, la première lettre du bigramme sur la première ligne et la seconde sur la deuxième. On prend ensuite 7 caractères (rappelons que 7 est l’entier faisant partie de la clé) de la première ligne, puis 7 de la seconde, puis encore 7 de la première etc.

**Exemple** : pour chiffrer « jumping jack flash» on obtient tout d’abord

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| clair | j | u | m | p | i | n | g |  | j | a | c | k | f | l | a |  | s | h |
| 1ère | J | D | S | T | D | T | J |  | J | S | D | T | J | L | S |  | S | L |
| 2ème | J | G | R | O | O | G | G |  | J | G | R | F | F | J | G |  | J | G |

D’où le chiffrement « JDSTDTJ JGROOGG JSDTJLS JGRFFJG SLJG » Que l’on peut remettre en forme en utilisant les conventions usuelles : «JDSTD TJJGR OOGGJ SDTJL SJGRF FJGSL JG ».

Le déchiffrement s’obtient en appliquant ces règles « à l’envers » : on reconstitue chacune des deux lignes, puis chacun des bigrammes, que l’on convertit enfin en lettres.

## Implémentation en Python

**Il vous est fortement recommandé de lire l’intégralité de cette partie avant de commencer à coder. Les travaux demandés sont mis en évidence avec une couleur bleue.**

**Dans un fichier que l’on nommera « chiffreDeCollon.py », implémenter les sous-programmes suivants :**

* Une fonction prenant en paramètre une chaîne de caractère, et qui retourne une liste à deux dimensions de 5 lignes et 5 colonnes, remplie ligne par ligne à l’aide de la chaîne passée en paramètre. On supprimera les doublons éventuels, et si la chaîne est “trop courte“ on complètera le carré à l’aide des lettres manquantes de l’alphabet prises dans l’ordre. Par exemple, si on passe « service académique » en paramètre, on obtient le carré :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| s | e | r | v | i |
| c | a | d | m | q |
| u | b | f | g | h |
| j | k | l | n | o |
| p | t | x | y | z |

* Une fonction prenant en paramètre une lettre et une clé (i.e. une liste de la forme présentée à la partie 2.2), et qui retourne la lettre chiffrée sous forme d’un bigramme selon l’algorithme de Collon.
* Une fonction prenant en paramètre un texte et une clé (i.e. une liste de la forme présentée à la partie 2.2 et un entier), et qui retourne le texte chiffré selon l’algorithme de Collon. On le mettra en forme en respectant les conventions habituelles.
* Une fonction prenant en paramètre un bigramme et une clé (i.e. une liste de la forme présentée à la partie 2.2), et qui retourne la lettre obtenue en déchiffrant le bigramme selon l’algorithme de Collon.
* Une fonction prenant en paramètre un texte et une clé (i.e. une liste de la forme présentée à la partie 2.2 et un entier), et qui retourne le texte déchiffré selon l’algorithme de Collon.

**Utilisation de ces procédures et fonctions :**

* Déchiffrer le texte ‘chiffre3.txt’ avec la clé générée à partir du mot « service académique » et l’entier 7. On accèdera aux données de ce fichier texte avec les fonctions usuelles de lecture, et non via un copier/coller dans le script.
* Chiffrer un texte de votre choix avec une clé de votre choix également et joignez la clé et le texte chiffré à votre projet sous la forme de fichiers “.txt“.

**Deux petites questions :**

* Que pensez-vous de la sécurité de cet algorithme ?
* Quels sont ses principales qualités et défauts ?