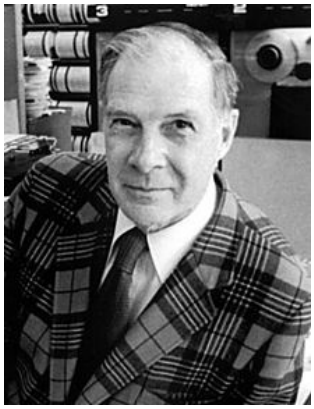
1re partie : TRANSMISSION FIABLE DES DONNEES

La transmission d'informations peut-être sujette à des perturbations :

- les téléphones cellulaires sont mobiles, relativement peu puissants, et souvent utilisés soit loin des antennes relais, soit dans un environnement urbain très bruyant du point de vue électromagnétique;
- les sondes spatiales n'ont pas à leur disposition d'énormes quantités d'énergie pour émettre des messages, se trouvent à des distances astronomiques, et leur antenne, même si elle est orientée le mieux possible, n'est pas parfaite;
- en cas de conflit armé, les communications adverses sont une des cibles privilégiées pour le brouillage et la guerre électronique.

Le stockage des informations peut lui aussi être sujet à des altérations, qu'il convient de détecter et corriger.

1.1. Origine des codes détecteur d'erreur



Depuis 1946, Richard Hamming (1915-1998) travaillait sur un modèle de calculateur à carte perforée de faible fiabilité. Si, durant la semaine, les ingénieurs présents pouvaient corriger les erreurs, les machines s'arrêtaient de manière quasi-systématique pendant les périodes chômées à cause de « bugs ».

Richard Hamming

Claude Elwood Shannon

Hamming a donc développé un code correcteur qui constitue les prémisses de la théorie des codes. La période de l'après-guerre correspond à la naissance de la théorie de l'information. Claude Shannon (1916-2001) formalise cette théorie comme une branche des mathématiques.

1.2. Principe de certains codes de détection d'erreur

L'exemple le plus simple consiste à coder une transmission sur un certain nombre de bits, auxquels s'ajoute un bit appelé bit de parité. C'est notamment le cas avec les transmissions série de type RS232 dans ce cas, le protocole comprend généralement :

- 1 bit de départ ;
- 7 à 8 bit de données ;
- 1 bit de parité optionnel ;
- 1 ou plusieurs bits d'arrêt.

Ce bit de parité peut, par exemple prendre la valeur 0 si le nombre de 1 contenu dans le message de 7 bits est pair, 1 si le nombre de 1 contenu dans le message est impair (une convention inverse peut aussi être retenue).

Attention, il ne faut pas confondre parité d'un nombre au sens mathématique et celle définie ici : 0000011 correspond au nombre 3 en base 10, nombre impair car non divisible par 2. La parité retenue pour la détection d'erreur est quant à elle paire (2 bits prennent la valeur 1 dans le mot de 7 bits).

Les détections d'erreur utilisant un unique bit de parité sont cependant limitées : si deux erreurs se produisent, leurs effets peuvent s'annihiler. De même, une réorganisation des 7 bits du message ne sera pas détectée comme une erreur. Par ailleurs, lorsqu'une erreur est détectée, il est impossible de savoir quel bit a été altéré.

Plusieurs sommes de contrôle (*check sum* en anglais) sont alors nécessaires pour détecter et corriger les erreurs.

2. Contrôle des données : parité croisée

Le contrôle de parité croisée est notamment bien adapté au contrôle des données stockées sur bande magnétique lorsqu'au moins deux têtes de lecture opèrent. Il peut aussi être utilisé dans la transmission de données.

On considère alors les valeurs contenues dans un rectangle d'une bande. Ces données peuvent alors être représentées ainsi :

0	1	1	0	0	1	0
0	1	0	0	0	0	0
1	1	0	1	1	1	1
0	0	0	0	1	1	0
1	1	0	0	0	1	1
1	1	0	0	0	0	1
0	1	0	1	1	0	0

Le contrôle de parité consiste à ajouter un bit de parité à la fin de chaque ligne et en bas de chaque colonne. Le stockage prend alors la forme :

0	1	1	0	0	1	0	1
0	1	0	0	0	0	0	1
1	1	0	1	1	1	1	0
0	0	0	0	1	1	0	0
1	1	0	0	0	1	1	0
1	1	0	0	0	0	1	1
0	1	0	1	1	0	0	1
1	0	1	0	1	0	1	0

Si une valeur de la donnée codée est altérée, l'erreur sera détectée en fin de ligne et en bas de la colonne correspondante. Si un bit de parité est altéré, une seule erreur sera détectée.

Le code de parité croisée peut aussi être mis en œuvre dans le cas de la transmission de données (voir exercice).

3. *Code de Hamming*

Un code de Hamming est un code correcteur linéaire. Il permet la détection et la correction automatique d'une erreur si elle ne porte que sur une lettre du message.

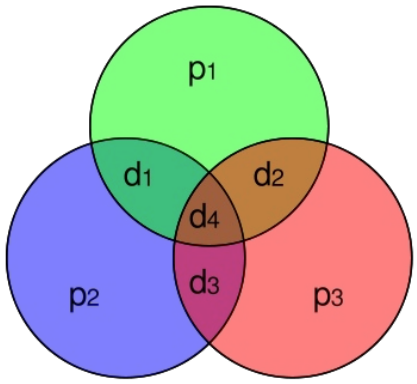
Il s'agit d'un code parfait : pour une longueur de code donnée il n'existe pas d'autre code plus compact ayant la même capacité de correction. En ce sens son rendement est maximal.

Il existe une famille de codes de Hamming ; le plus célèbre et un des plus simple est le code binaire de paramètres [7,4] (c'est aussi celui auquel le programme des concours fait référence). À travers un message de 7 bits, il transfère 4 bits de données et 3 bits de parité.

La figure ci-contre est une représentation graphique de ce code.

Le message à transmettre est constitué des bits d_1, d_2, d_3 et d_4.

Le message complet est constitué de trois sommes de contrôles p_1, p_2, p_3, et des quatre lettres du message initial. La valeur de p_i est égal à 0 si la somme des trois lettres du message incluses dans son cercle sur la figure est paire et 1 sinon.



Si une altération se produit, par exemple sur p_1 alors la parité du cercle de p_1 est modifiée ; en revanche celles des cercles de p_2 et de p_3 ne sont pas modifiées. Si la parité de d_1 est modifiée, alors celles des cercles de p_1 et de p_2 le sont mais celle de p_3 ne l'est pas.

Le tableau ci-dessous récapitule les différentes altérations ainsi que leurs effets sur les parités.

bit	**1**	**2**	**3**	**4**	**5**	**6**	**7**
bit altéré	p_1	p_2	d_1	p_3	d_2	d_3	d_4
Cercle « p3 »	OK	OK	OK	modifiée	modifiée	modifiée	modifiée
Cercle « p2 »	OK	modifiée	modifiée	OK	OK	modifiée	modifiée
Cercle « p1 »	modifiée	OK	modifiée	OK	modifiée	OK	modifiée

L'ordre choisi pour les différents bits n'est pas anodin : si l'on transforme les « OK » en 0 et les « modifiée » en 1, on obtient :

bit	**1**	**2**	**3**	**4**	**5**	**6**	**7**
bit altéré	p_1	p_2	d_1	p_3	d_2	d_3	d_4
Cercle « p3 »	0	0	0	1	1	1	1
Cercle « p2 »	0	1	1	0	0	1	1
Cercle « p1 »	1	0	1	0	1	0	1
Nombre binaire	001	010	011	100	101	110	111
En base 10	1	2	3	4	5	6	7

En reprenant les chiffres binaires, on obtient la suite des nombres de un à sept en binaire. Cette propriété permet par la suite un décodage aisé.

Encodage d'un message

L'ensemble E des messages à envoyer est celui de mots de quatre « lettres » prises dans l'ensemble {0,1} ; le message est codé en un mot de sept lettres prises dans le même ensemble. On note F l'espace des mots de sept lettres binaires. E et F peuvent être considérés comme des espaces vectoriels sur le corps binaire {0,1}.

L'encodage, c'est-à-dire l'opération consistant à transformer le message de E de quatre lettres en un code de F de sept lettres apparaît alors comme une application linéaire de E dans F. Elle se décrit par une matrice.

L'encodage consiste alors à multiplier le vecteur de quatre lettres binaires par une matrice 7x4 pour obtenir un vecteur composé de sept lettres binaires.

La connaissance de l'image de chaque vecteur de la base canonique détermine entièrement l'application d'encodage φ.

Les quatre vecteurs de la base correspondent aux messages suivants : d_1 = 1000, d_2 = 0100, d_3 = 0010 et d_4 = 0001.

L'image de d_1, le message qui a des coordonnées nulles en d_2, d_3 et d_4, possède deux parités égales à un, celle de p_1 et p_2 et une égale à zéro : p_3.

En respectant l'ordre de la base de F : p_1, p_2, d_1, p_3, d_2, d_3 et d_4, on obtient l'image du premier vecteur :

$$\varphi(1000) = 1110000$$

De la même manière, on obtient les images des autres vecteurs de la base de E :

$$\varphi(0100) = 1001100\,, \quad \varphi(0010) = 0101010\,et \quad \varphi(0001) = 1101001$$

La matrice génératrice du code est formée des quatre colonnes correspondant aux images des vecteurs de la base canonique par φ, on obtient :

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Le codage d'un message de 4 bits résulte alors de la parité de chaque composante obtenue en effectuant le produit de la matrice génératrice par le vecteur contenant le message initial.

Décodage d'un message

Le principe est similaire à celui adopté pour l'encodage. Une matrice de contrôle H peut être définie à partir du tableau regroupant les différentes altérations. Trois conditions (sur p1, p2 et p3) doivent être remplies pour garantir que le message a été transmis sans altération. La matrice de contrôle H contient donc 3 lignes. Chaque condition s'exprime comme une somme devant être paire, ou encore nulle dans le corps binaire.

Le produit de la matrice de contrôle H par le message (4 bits + 3 bits de contrôle) donne le vecteur nul si le message n'a pas été altéré.

La matrice de contrôle est la suivante :

$$\mathbf{H} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

En cas d'erreur, le vecteur résultant du produit de la matrice de contôle par le message donne un vecteur non nul dont les trois composantes peuvent être converties en nombre binaire. En reprenant le tableau récapitulatif des erreurs, on peut alors retrouver le bit qui a été alterré.

Limite du code de Hamming [7,4]

En cas d'une double erreur, le code détecte une anomalie. La correction apportée est alors erronée puisque le code d'erreur ne correspond pas aux erreurs. Pour remédier à ce problème, un huitième bit peut être ajouté. Ce huitième bit correspond à la parité de l'ensemble du message (p1, p2, d1, p3, d2, d3, d4). Le code de Hamming utilisé est alors un code de Hamming [8, 4].

4. Exercice 1 : mise en œuvre d'un code de détection d'erreur par parité croisée (d'après un dm de M. Médevielle)

On considère le message initial codé de la manière suivante :

[**0,1,1,1,1,0,0**,1,1,1,1,1,1,1,**1,0,0,0,0,1,1**,1,1,0,0,0,0,0,**0,0,0,0,0,0,1**,1,1,0,1,1,0,1,**1,1,0,0,0,0,0**]

Ce message est formé d'une trame de 49 bits.

1) Écrire une fonction qui reçoit le message et qui le met en forme de manière à ce qu'il soit stocké sous la forme d'une liste contenant n listes (n = 7 par défaut).
2) Écrire une fonction qui reçoit une liste et qui ajoute à cette liste son bit de parité (0 correspond à un nombre pair de 1).
3) Écrire une fonction qui reçoit une liste de listes de même taille et qui ajoute une liste contenant les bits de parité (même convention que pour les lignes) correspondant à chaque colonne.
4) Écrire une fonction qui reçoit un message sous la forme d'une liste et qui le code sous la forme d'un tableau de n + 1 lignes contenant le message et les bits de parité.
5) Écrire une fonction qui reçoit un message contenant une unique erreur au sein du message initial et qui retourne les coordonnées du bit erroné.
6) Écrire une fonction qui reçoit un message, utilise la fonction précédente et retourne le message corrigé ou le message initial s'il n'y a aucun bit altéré.

5. Exercice 2 : mise en œuvre d'un code de Hamming [7,4]

À partir de la description du code Hamming [7, 4], écrire les trois fonctions suivantes :

- Fonction **codage** : elle reçoit en entrée une liste comprenant 4 valeurs (0 ou 1) correspondant au message à coder et retourne le message codé.
- Fonction **decodage** : elle reçoit en entrée une liste de 7 valeurs correspondant à un message codé. Elle retourne un booléen (True si le message ne comprend pas d'erreur, False dans le cas contraire) et la liste comprenant les valeurs issues du produit de la matrice de contrôle par le vecteur contenant le message complet.
- Fonction **correction** : elle reçoit en entrée une liste de 7 valeurs correspondant au message complet. Elle utilise les résultats de la fonction **decodage**, corrige les éventuelles erreurs et retourne le message de 4 bits initial sous forme d'une liste.

2de partie : CRYPTOGRAPHIE

Le mot cryptographie vient des mots en grec ancien kruptos (« caché ») et graphein (« écrire »). La cryptographie est une des disciplines de la cryptologie s'attachant à protéger des messages (assurant confidentialité, authenticité et intégrité) en s'aidant souvent de secrets ou clés. Utilisée depuis l'antiquité, l'une des utilisations les plus célèbres pour cette époque est le chiffre de César, nommé en référence à Jules César qui l'utilisait pour ses communications secrètes.

6. *Exercice 3 : La scytale*

Chez les Spartiates, la scytale, également connue sous le nom de bâton de Plutarque, était un bâton de bois utilisé pour lire ou écrire une dépêche chiffrée. Considérée comme le plus ancien dispositif de cryptographie militaire connue, elle permettait l'inscription d'un message chiffré sur une fine lanière de cuir ou de parchemin. Après avoir enroulé la ceinture sur la scytale, le message était écrit en plaçant une lettre sur chaque circonvolution. Pour le déchiffrer, le destinataire devait posséder un bâton d'un diamètre identique à celui utilisé pour l'encodage. Il lui suffit d'enrouler la scytale autour de ce bâton pour obtenir le message en clair.

1) Écrire une fonction qui permette d'encoder un message passé en argument en fonction d'un nombre de faces de la scytale (la clé).
2) Écrire la fonction qui permet le décodage du message à partir de la clé.

7. *Exercice 4 : Le code César*

Postérieur et plus simple que la scytale, le chiffre de César doit son nom à Jules César qui l'utilisait pour certaines de ses correspondances secrètes, notamment militaires.

Il s'agit d'un cryptage par substitution : une lettre de l'alphabet est remplacée par une autre grâce à un décalage de d lettres vers la droite ou vers la gauche. Le nombre d est alors appelé clé.

On rappelle les deux fonctions suivantes en **Python** :

- **ord('c')** renvoie 99, le code ASCII de la lettre 'c'.
- **chr(99)** effectue l'opération inverse.

1) Écrire une fonction qui permette d'encoder un message passé en argument en fonction du décalage d (la clé).
2) Écrire la fonction qui permet le décodage du message à partir de la clé.

8. Exercice 5 : Le chiffre Vigenère

Le chiffre de Vigenère est un chiffrement basé sur une substitution dont la clé est constituée par un texte : une lettre de l'alphabet dans le texte en clair peut être chiffrée de plusieurs manières.

Au XVIème siècle, Blaise de Vigenère fut l'un des premiers à présenter ce type de chiffrement sous la forme d'une table avec la présence d'une clé secrète.

Le chiffre de Vigenère est resté inviolé pendant plusieurs siècles.

Blaise de Vigenère, 1523 -1596, est un diplomate et cryptographe français.

Si l'on prend par exemple la clé « concours ». Pour crypter un texte, on code de la même manière qu'avec le code César la première lettre en utilisant le décalage qui remplace la lettre a par la lettre c, première lettre de la clé. Pour la deuxième lettre, on utilise le code qui remplace la lettre a par la lettre o. Et ainsi de suite. Pour la neuvième lettre, on reprend la clé correspondant à la première lettre.

1) Écrire une fonction qui permette d'encoder un message passé en argument en fonction d'une clé donnée sous la forme d'un texte.
2) Écrire la fonction qui permet le décodage du message à partir de la clé.