# TD de théorie de l'information

## Année 2008-09

## Version 1.1

Université de Montpellier
Place Eugène Bataillon
34095 Montpellier Cedex 5

Rodolphe Giroudeau
161, rue Ada
34392 MONTPELLIER Cedex 5
TEL : 04-67-41-85-40
Mail : rgirou@lirmm.fr

Université de Montpellier II
MasterM2

2009/2010
Durée: 12h

**T**héorie de l'information
**T**D – Séance n° 1

---

# 1 Mesure de l'information

**Exercice 1 – Entropie et codage**

Une séquence expérimentale d'ADN est définie comme une suite de smboles certains ou incertains.Ùn symbole certain est un élément de l'alphabet : $B = \{A, T, G, C\}$. Un symbole peut être incertain d'ordre 2 ou incertain d'ordre 3. Un symbole incertain d'ordre 2 est un élément de l'ensemble des arrangements de 2 éléments de $B$ ; il y a donc un ordre interprété comme le fait que la première lettre de l'arrangement est plus probable que la seconde. Par exemple : $TA$ est un symbole incertain d'ordre 2 ; $AT$ en est un autre. Un symbole incertain d'ordre 3 est un élément de l'ensemble des combinaisons de 3 éléments de $B$ ; il n'y a donc par d'ordre. Par exemple $TGA$ ou $AGT$ ou $TAG$ etc ... sont des notations d'un même symbole incertain d'ordre 3.

Une série d'expériences a montré les résultats suivants sur un ensemble de séquences :

1. les symboles certains sotn équiprobables entre eux
2. tous les symboles incertains (ordre 2 et 3 confondus) sont aussi équiprobables entre eux
3. la probabilité d'un symbole certain dans une séquence est la même que celle d'un symbole incertain.

1. Calculer l'informaiton d'un symbole certain et l'information d'un symbole incertain dans une séquence expérimentale.
2. Caucler l'entropie d'une séquence exprérimentale.
3. Trouver un codge optimal pour chaque symbole d'une séquence expérimentale.
4. Un généome humain est une suite de $3,5$ milliards de symboles. En supposant cette suite composée de symboles indépendants, et en se plaçant dans le même cadre expérimentale, que doit on prévoir comme capacité de stockage sur un support exempt d'erreurs pour archiver la séquence expérimentale d'un génôme ?

**Exercice 2 – Entropie et codage suite**

1. Calculer l'entropie $H_x$ d'une source binaire $X = \{0, 1\}$ telle que $\mathbb{P}(x = 0) = p$ et $\mathbb{P}(x = 1) = 1 - p$.
2. Etudier et interpréter $H_x(p)$.

## 1.1 Capacité d'un canal

**Exercice 3 – Canal dissymétrique**



On considère un canal binaire, dissymétrique, stationnaire et sans mémoire. La transmission d'un bit égal à 0 se fait sans erreur. La transmission d'un bit égal à 1 se fait avec une erreur aléatoire de 50%

1. Calculer l'équivoque et la transinformation pour une source équiprobable.
2. Calculer la capacité de ce canal.

**Exercice 4 – Canal à « effacement »**

On considère un canal binaire symétrique snas erreur, mais avec pertes détectées par un mécanisme approprié. L'alphabet de réception comporte donc 3 éléments : $0, 1$ et $\epsilon$ qui désigne le bit détecté « perdu ». Soit $p$ la probabilité de perte d'un bit.

1. Calculer l'équivoque et la transinformation pour une source équiprobable.
2. Calculer la capacité de ce canal pour $p = 1/2$.

**Exercice 5 – Transmission bruitée**

On s'intéresse à un nouveau standard de Télévision $TBD$ : Ttrès Basse Définition pour des applications militaires. Une image est composée de pixels. Chaque pixel est défini par de l'une des trois couleurs : $R$ $V$ et $B$. On a mesuré sur une grande série d'images la répartition en probabilités des trois couleurs :

$$R : 1/4 \; V : 1/2 \; B : 1/4$$

1. Première partie
   (a) Quelle est la quantité d'information asscoiée à chacune des 3 couleurs.
   (b) Quelle est l'entropie de la source ?
   (c) Donner le codage optiaml d'un pixel.
2. La transmission a lieu dans un environnement très bruité. LA matrice de transmission est ainsi définie : soit $X$ le symbole émis et $Y$ le symbole rçu, appartenant chacun à l'alphabet $\{R, V, B\}$. $\mathbb{P}(Y|X) = 1/2$ si $Y = X$, $1/4$ sinon.
   (a) Quelle est la matrice de réception $\mathbb{P}(X|Y)$ donnant pour chaque symbole reçu la probabilité du symbole émis ?
   (b) Calculer l'entropie de la source de chrominance après réception : $H(X|Y)$.
   (c) Calculer la quantité d'information transmise $I(X;Y)$.

On donne $\log_2 3 = 1,58, \log_2 5 = 2,32$ et $\log_2 7 = 2,81$

**Exercice 6 – Tranmission bruitée suite**

On considère une source binaire symétrique $X = \{0, 1\}$, donc $p = 1/2$ émettant sur un canal bruité symétrique, stationnaire et sans mémoire vers un puits également et symétrique $Y = \{0, 1\}$. La matrice de bruit $B_{Y|X}(q)$ est définie par la seule quantité $q$ telle que :
– $\mathbb{P}(y = 1|x = 0) = \mathbb{P}(y = 0|x = 1) = q$
– $\mathbb{P}(y = 0|x = 0) = \mathbb{P}(y = 1|x = 1) = 1 - q$

1. Calculer les quantités :
   (a) $\mathbb{P}(x = 0|y = 0)$

(b) $\mathbb{P}(x = 0|y = 1)$
(c) $\mathbb{P}(x = 1|y = 0)$
(d) $\mathbb{P}(x = 1|y = 1)$
(e) $\mathbb{P}(y = 0)$
(f) $\mathbb{P}(y = 1)$

2. En déduire $H_Y, H_{X|Y}$ et $I_{X;Y}$ en fonction de $q$.
3. Etudier et interpréter $I_{X;Y}(q)$.

## 2 Compression

**Exercice 7 – Sur l'existence d'un compresseur universel**

1. Donner le nombre de mots pour coder un mot de longueur $n$ sur l'alphabet $\{0, 1\}$ ?
2. Donner le nombre de mots pour coder un mot de longueur au plus $n-1$ sur l'alphabet $\{0, 1\}$ ?
3. Conclure.

**Exercice 8 – Code préfixe**

Soit $S$la source d'alphabet $\{a, b, c, d\}$ et de probilité $P(a) = 0,5, P(b) = 0,25, P(c) = 0,125, P(d) = 0,125$. On code $S$ avec le code suivant : $a = 0, b = 10, c = 110, d = 111$.

1. Calculer *adbccab*. Décoder 1001101010.
2. Est-ce un code instantané ?
3. Caculer l'entropie de la source.

**Exercice 9 – Compression**

On veut construire un code compresseur binaire sur une source $\mathcal{S} = (S, P)$. (supposée infinie) où $S = (0, 1)$ est l'alphabet source $P = (P(0) = p, P(1) = 1 - p)$ est la loi de probabilité de $\mathcal{S}$.

1.

**Exercice 10 – Compression**

1. Compresser le mot ci-dessus avec la méthode « Run Lenght Coding » $AAAABBCCCBBCCCAC$
2. Quelle est la taille du codage obtenu ?
3. Quelle est la condition structurelle permettant de garantir qu'un mot est compactable ?
4. Quelle est la condition fonctionnelle permettant de garanir qu'un mot est compactable ?

**Exercice 11 – Méthode topologique**

1. Compresser la source composé par deux mots suivants avec la méthode topologique : $AAAABBCC$ et $CBBCCCAC$.
2. Quelle est la taille du code obtenu ?
3. Quelle est la condition totpologique permettant de garantir qu'un mot est compactable ?
4. Peut-on recompresser, c'est à dire appliquer à nouveau la méthode topologique sur la nouvelle source ontenue ?



**Exercice 12 – Méthode RLE**

1. Quel serait le codage RLE de l'iamge donnée par la figure **??** ?
2. Y-a-t'il un gain ?
3. Quel serait le codage sur bloc de 5 bits en 16 niveaux de gris ? ET le gain ? Peut-on compresser plus avec ce système ?
4. Et en 255 niveaux de gris (avec au plus 16 pixels identiques consécutifs) ?
5. Et en colonnes ?

**Exercice 13 – Méthode Move-To-Front**

Soit $A$ l'alphabet sur 8 symboles suivant : $A = (a, b, c, d, m, n, o, p)$.

1. On associe à chaque symbole un numéro entre 0 et 7, selon leur position alphabétique. Quels nombres représentent « abcddcbamnopponm » et « abcdmnopabcdmnop » ?
2. Coder les deux chaînes précédentes en utilisant la méthode Move-To-Front. Que constatez-vous ?
3. Combien de bits sont nécessaires pour coder les deux premiers nombres, par Huffman par exemple ?
4. Combien de bits sont nécessaires pour coder les deux nombres obtenus après Move-To-Front ? Comparer les entropies.
5. que donne Huffman étendu à deux caractères sur le dernier nombre ? Quelle est alors la taille de la table des fréquences ?
6. Comment pourrait-on qualifier l'algorithme composé d'un Move-To-Front suivi d'un code statistique ?
7. Le Move-ahead-$k$ est une varaition du Move-To-Front où le caractère est seulement avancé de $k$ positions au lieu du sommet de la pile. Encoder le sdeux chaînes précédenetes avec $k = 1$ puis $k =$ et comparer les entropies.

**Exercice 14 – Analyse du code BWT**

1. La dernière colonne $L$ de l matrice triée de la transformation BWT contient des concentrations de caractères identiques, c'est pourquoi $L$ se compresse bien. Toutefois, la première colonne $F$ se compresserait encore mieux puisqu'elle est totalement triée. Pourquoi sélectionner $L$ plutôt que $F$ comme code ?
2. Soit la chaîne $S =<< ssssssssh >>$. Calculer $L$ et sa compression Move-To-Front.
3. Implémentation pratique : BWT est efficace sur de longues chaînes $S$ de taille $n$. En pratique, il est donc impensable de stocker toute la matrice $n \times n$ des permutations. En fait, il faut seulement trier ces permutations et pour les trier, il suffit d'être capable de comparer deux permutations.
   (a) Donner un indice qui soit capable de désigner et différencier les permutations de la chaîne de départ.
   (b) Construire un algorithme *comparerpermutations* calculant une fonction booléenne qui a pour entrée une chaîne $S$ et deux entiers $i$ et $j$. cet algorithme doit décider si la chaîne déclée $i$ fois est avant la chaîne décalée $j$ fois, dans l'ordre obtenu en triant toutes les permutations par ordre alphabétique.

(c) Conlure sur l'espace mémoire nécessaire pour calculer la BWT.

(d) Comment calculer $L$ et l'index prmaire à partir du tableau des permutations ?

**Exercice 15 – Algorithme de Shannon-Fano**

1. Compresser la source suivante avec l'algorithme de Shannon-Fano $AAAABBCCCBDDCAE$.
2. Quelle est la taille du code obtenu ?
3. Quelle est la taille de l'en-tête ?
4. Peut-on ré-applqiuer l'algorithme sur la nouvelle source obtenue (en-tête non comprise) ?

**Exercice 16 – Pile ou Face pour jouer au** 421

On souhaite jouer au lancé de dé, avec pour unique moyen une pièce de monnaie. On va donc chercher à coder un dé non pipé à 6 faces avec une pièce non piipée à deux faces.

1. Quelle est l'entropie d'un dé ?
2. Proposer un algorithme de codage.
3. Calculer la longueur moyenne de ce codage.
4. Ce codage est-il optimal ?

**Exercice 17 – Algorithme de Huffman : aspect théorique**

Cet exercice introduit des éléments théoriques sur la valeur du code généré par l'algorithme de Huffman. Soit $\mathcal{S} = (S, P)$, où $S = (0, 1)$, et $P(0) = 0,99$ et $P(1) = 0,01$.

1. Calculer l'entropie de $\mathcal{S}$.
2. Donner le code généré par l'algorithme de Huffman sur la troisième extension $\mathcal{S}^3$. Quel est le taux de compression ?
3. Que pouvez-vous dire de l'optimalité de l'algoeirhmz de Huffman en comparant les taux de compression obtenus avec ceux de l'exercice précedent ? Cela est-il conformae au théorème de Shannon ?

**Exercice 18 – Algorithme de Huffman**

On cherche à coder des jets successifs (en nombre supposé infini) d'un dé faussé. Les symboles de la source sont notés $(1, 2, 3, 4, 5, 6)$, et suivent la loi de probabilité d'apparition $(0,12; 0,15; 0,16; 0,17; 0,18; 0,22)$.

1. Quelle est l'entropie de cette source ?
2. Proposer un code ternaire (sur un alphabet à trois chiffres) issu de l'algorithme de Huffman pour cette source. Quelle est la sa longueur moyenne ? Quelle longueur moyenne minimale peut-on espérer pour un tel code ternaire ?
3. Même question pour un code binaire

**Exercice 19 – Suite de l'exercice précédent**

L'algorithme de Huffman construit un code où des mots de osurce de longueur fixes sont codés par des mots de code de longueur variable. L'organisation de la mémoire où l'on stocke le code de longueur fixe aux mots de code, et on code alors des séquences de longueur variable des chiffres de la source.

Les codes dits de Tunstall sont des codes optimaux qui suivent ce principe. En voici la méthode de construction dasn le cas d'un alphabet binaire. Si la longueur choisie des mots de code est $k$, il faut choisir les $2^k$ séquences de chiffres de source qu'on va choisir de coder. Au départ, l'ensemble des candidats est l'ensemble des mots de longueur 1 (ici $\{1, 2, 3, 4, 5, 6\}$). Soit dans cet ensemble le mot le plus probable (ici, 6). On construit toutes les séquences réalisable en rajoutant un chiffre à ce mot, et on remplace ce mot par ces séquences dans l'ensemble des candidats (ici on obtient alors $\{1, 2, 3, 4, 5, 61, 62, 63, 64, 65, 66\}$). On remmonce alors cette opération tant que la taille de l'ensemble des candidats n'est pas strictement supérieure à $2^k$ (on arrête avant d'avoir dépassé cette valeur). On code alors toutes ce séquences par des mots de longueur $k$.

1. Sur un alphabet binaire, construire un code de Tunstall pour le dé, pour des longueurs de mots de code $k = 4$.
2. Comment est codée la séquence « 6664 » par Tunstall ? Par Huffman ?
3. Pour chaque mot de code, calculer le nombre de bits de code nécessaires par caractère de source. En déduire la longueur moyenne par bit de source code de Tunstall.

**Exercice 20 – Algorithme de Tunstall**

Nous poursuivons l'étude l'algorithme de Tunstall. Soit $\mathcal{S} = (S, P)$, où $S = (A, B, C)$, et $P(A) = 0,6$, $P(B) = 0,3$ et $P(C) = 0,1$.

1. Quelle est la relation entre $N$ le nombre d'élements dans $S$, $K$ le nombre d'itérations et $k$.
2. Donner un 2-bit code tele que le décodage de la séquence suivante soit impossible $AAABAABAABAABAAA$.
3. Donner un 3-bit code de Tunstall pour la source décrite ci-avant.
4. Coder $AAABABCBAA$.

**Exercice 21 – Algorithme de Huffman**

Considérons la source suivante : « Le president est entre dans la salle » Nous avons supprimé les accents, mais les espaces sont comptés.

1. Construire la table des fréquences.
2. Construire l'arbre de Huffman.
3. Compresser la source avec le codage Huffman.
4. Calculer la taille de l'en-tête.
5. Quel est le gain de compression ?

**Exercice 22 – Compression par Huffman**

Une source emet des symboles de l'alphabet $A = \{a, b, c, d, e\}$ avec les probabiltés :

- $\mathbb{P}(a) = 0,15$
- $\mathbb{P}(b) = 0,04$
- $\mathbb{P}(c) = 0,26$
- $\mathbb{P}(d) = 0,05$
- $\mathbb{P}(e) = 0,5$

1. Calculer l'entropie de cette source.
2. Trouver un code de Huffman pour cette source.
3. Calculer la longueur moyenne du code de Huffman.

**Exercice 23 – Algorithme de Huffman**

Considérons la source suivante : $GATTACA$ répété deux fois.

1. Construire la table des fréquences.
2. Calculer l'entropie de la source.
3. Compresser la source avec le codage Huffman.
4. Quelle est la taille du code compressé (sans l'entête) ?
5. Quelles sont les tailles obtenues si le motif est répété $5, 10, 20$ fois ? Calculer le gain avec et sans entête avec les répétitions.

**Exercice 24** – **Huffman dynamique**

Nous considérons la séquence *aaa aaa aaa bcd bcd bcd*.

1. Quels erait le codage de Huffman dynamique de cette séquence ?
2. Quel serait le codage de Huffman de l'enxtension à trois caractères de cetee séquence ?

**Exercice 25** – **Compression par LZ78**

Compresser la séquence suivante $(ab)^{16}$ avec LZ78

**Exercice 26** – **Compression par LZ78**

Reprendre l'exemple du cours avec les paramètres $N = 12$ et $F = 5$.

**Exercice 27** – **Compression par LZ78**

Considérons la source suivante : « MAMAN AIME LES MANGUES » Nous avons supprimé les accents, mais les espaces sont comptés.

1. Compresser la source avec le codage $LZ78$.
2. Construire l'arbre de décompression.
3. Quel est le gain de compression ?

**Exercice 28** – **Compression suite**

Considérons la source composé par la motif « ATGC» répété deux fois.

1. Calculer l'entropie de la source.
2. Compresser la source avec le codage $LZ78$.
3. Quelle est la taille du code compressé ?
4. Quelles sont les tailles obtenues si le motif est répété 3 , 4 5 fois ? Donner les arbres de décompression pour chaque cas.

**Exercice 29** – **Compression**

Une source emet des symboles de l'alphabet $A = \{a, b, c\}$ avec les probabiltés :

- $\mathbb{P}(b|a) = 0,5$
- $\mathbb{P}(c|b) = 0,5$
- $\mathbb{P}(a|b) = 1$
- $\mathbb{P}(a|c) = 0,5$
- $\mathbb{P}(b|c) = 0,5$

La source émet 4 symboles et commence par émettre « a ».

1. Quelles sont les équences possibles émises par la source ?
2. Calculer la longueur d'un code $LZ$ obtenu pour chaque séquence.



3. En déduire la longueru moyenne du codage $LZ$ pour cete source.

**Exercice 30** – **Compression fin**

Alice, Bob et Dominique sont co-locataires d'un grand appartement. Ils mettent en commun leur importante collection de CD et cherchent un moyen de coder chacun de leur CD de telle sorte qu'il soit facile de toruver à qui appartient le CD et de quelle genre il est parmi : Classique, Jazz, Variétés.

- Alice possède 256 CD : 1/2 de classique, 1/4 de jazz et 1/4 de Variétés.
- Bob possède 512 CD : 1/4 de classique, 1/4 de jazz et 1/2 de Variétés.
- Diminique possède 256 CD : 1/4 de classique, 1/2 de jazz et 1/4 de Variétés.

Nous notons ainsi la collections :$A\ Alice,\ B BOB, D\ Dominique, C\ classique, J\ Jazz, V\ varits, \Gamma\ letout$

Chaque CD est représenté par deux lettres : le nom de la personne suivi du genre. Par exemple $AC$ est un CD Classique de Alice.

1. Calculer l'entropie de la collection $\Gamma$.
2. Trouver un codage de Huffman pour la collection $\Gamma$.

**Exercice 31** – **Compression**

Une source émet un ensemble de symboles $\mathcal{A} = \{a, b, c, d\}$ avec les probabilités : $p(a) = 0,5, p(b) = 0,25, p(c) = 0,2, p(d) = 0,05$.

Nous supposerons que le $a$ apparaît en une longue série de taille $n$. Les autres lettres apparaissent aléatoirement.

1. Calculer l'entropie de cette source
2. Trouver un code de Huffman pour cette source.
3. Si l'on code la source avec NRZ puis Huffman, pour quelle valeur de $n$ obtient-on un gain de compression ?

## 3 Codes correcteurs et correcteurs d'erreurs

**Exercice 32** – **Mesures des erreurs de transmissions**

On a mesuré que le serreurs de transmission sur une boucle locale hertzienne pouvaient être modélisées par un canal binaire symétrique et sans mémoire, avec une probabilité d'erreur dee 1/1000 sur chaque bit transmis. On transmet sur ce canal des trames d'une longueur de 1000 bits. Sachant que les erreurs sont indépedantes pour chaque bit :

1. Calculer le taux d'erreur d'une trame (probabilité d'au moins un bit erroné). $ln(1 + \epsilon) \equiv \epsilon$
2. Calculer le taux résiduel d'erreur après utilisation d'un code de parité simple. L'efficacité du code de parité sera calculée en faisant le quotient de l aprobabilité de détection d'une seule erreur sur la probabilié d'occurrences d'une ou deux erreurs.

**Exercice 33** – **Construction empirique d'un code correcteur**

On souhaite construire un code susceptible de corriger jusqu'à deux erreurs, pour une trame de deux bits.

1. Quelle doit être la distance minimum de ce code ?

Soit $k$ le nombre de bits à coder, $r$ le nombre de bits de contrôle à ajouter à $k$ pour la correction des erreurs et $n = k + r$. La théorie des codes de Hamming nous indique que pour pouvoir corriger une erreur, les $r$ bits de contrôle doivent pouvoir coder soit l'absenc d'erreur, soit la position de l'erreur quand il y en a une. Il faut donc que $r \geq \log_2(n+1)$.

2. En s'inspirant des codes de Haming, exprimer l'inéquation permettant de calculer la valeur minimum à donner à $r$ pour pouvoir corriger jusqu'à deux erreurs. On calculera pour cela le nombr de possibilités d'avoir $0, 1$ ou $2$ erreurs.
3. En utilisant les résultats des questions précédentes plus un peu d'imagination de d'I.N. (Intelligence Naturelle), cosntruire un cde éventuellment systématique, de longueur $n = 8$ bits pouvant corriger deux erreurs poiur un bloc à coder de dimension $k = 2$ bits. Justifier la cosntruction et le spropriétés de ce code.

**Exercice 34 – Codes linéaires**

On considère le code linéaire $C$ dont la base est formée par trois mots :
- 00111
- 10101
- 11011

1. Vérifier que ces mots forment bien une base.
2. Que se passe-t'il si nous rajoutons le vecteur $X_4 = 01110$ ?
3. Quelles sont les valeurs $k$ (dimension de $C$, $n$ longueur de $C$ et $R$ redondance de $C$ ?
4. Construire la matrice génératrice $G$ de $C$.
5. Construire les mots du code $C$.
6. Montrer qe la distance de ce code est égale à son poids.
7. En conservant le même code, construire la matrice $G'$ permettant un codage systématique (ou séparable). Expliquer sa construction.
8. Construire la matrice de vérification $H'$ correspondant à $G'$.
9. Calculer à nouveau à partir de $H'$ la distance $d(\mathbb{C})$
10. Est-ce un code de Hamming ? Justifier.
11. Le récepteur reçoit le $Y_1 = 11111$. Quel est son syndrome ? Quel mot à été émis ?
12. Même question avec le mot $Y_2 = 10011$ ?
13. Quelle corrélation y a t'il avec la distance du code ?
14. Combien de code permet-il de détecter et de corriger d'erreurs ?

**Exercice 35 – Codes polynomiaux**

On considère le code cyclique $C(7,4)$ engendré par le polynôme générateur $g(x) = x^3 + x + 1$.

1. Rappler le sens de $\mathbb{C}(\mathrm{n}, \mathrm{k})$.
2. Montrer que le polynôme générateur est bien celui d'un code cyclique.
3. Ce code peut-il détecter (justifier chacune des réponses. On pourra utiliser des théorèmes connus en les citant) :
   (a) toutes les erreurs ?
   (b) toutes les erreurs doubles ?
   (c) toutes les erreurs de poids impair ?
4. Soit $f(x) = x^3 + x$ un bloc à coder
   (a) Quel est le mot du code $c(x)$ correspondant ? On considère le polnôme $h(x)$ tel que $g(x).h(x) = x^7 + 1$ On appelle syndrome d'un polynôme $a(x)$ l'expression $S(a(x)) = (h(x).a(x)) \mod (x^7 + 1)$.
   (b) Montrer que l syndrome de $c(x)$ est nul.
5. Soit $e(x) = x^2(x^2 + x + 1)$ un polynôme d'erreur.
   (a) Quel est le polynôme $d(x)$ reçu pour $c(x)$ transmis avec l'erreur $e(x)$ ?
   (b) Quel est le syndrome de $d(x)$ ?
   (c) L'erreur $e(x)$ est elle détectable ? Justifier.
6. Le code engendré par $g(x)$ par multiplication polynomiale est-il systématique ? Pourquoi ?
7. COmment engendrer un code systématique à partir de $g(x)$ ?
8. Quel est, dans ce cas, le mot de code $c'(x)$ engendré pour le bloc à coder $f(x)$ précédent ?
9. Peut-on détecter l'erreur $e(x)$ précdente ? Si oui comment ? Justifier.

**Exercice 36 – Codes linéiares**

On considère le code suivant :

$$C = \{00000000, 00011111, 11111000, 11100111\}$$

1. Calculer
   (a) Sa longueur $n$
   (b) Sa dimension $k$
   (c) Sa redondance $R$
   (d) Sa distance $D$
   (e) Le nombre d'erreurs détectables $Ed$
   (f) Le nombre d'erreurs corrigeables $Ec$
2. Montrer que ce code est linéaire. Construire une matrice génératrice $G$
3. Montrer que ce code n ?est pas systématique. Construire une matrice génératrice $G$ d'un code systématique équivalent. Construire la matrice de vérification $H$ correspondante
4. Le mot 11 est émis puis codé à partir du code systématique de matrice $G$? Une malencontreuse série de 2 erreurs transforme ces 2 bits en 00 Ces erreurs sont-elles détectables ? Justifier Ces erreurs sont-elles corrigeables ? Sinon pourquoi ? Si oui comment ?

**Exercice 37 – Sur les codes cycliques linéaires**

Montrer tous les codes cycliques linéaires possibles de longueur 6. Indiquer pour chacun d ?eux leurs caractéristiques essentielles : polynôme générateur, dimension, matrice de codage, distance, capacité de détections d ?erreurs. Quels sont, parmi ces codes, ceux permettant de détecter une erreur double sur 2 bits consécutifs, ceux permettant de détecter une erreur quadruple sur 4 bits consécutifs ?

Mêmes questions avec des codes de longueur 8, puis 5.

**Exercice 38 – Sur les codes cycliques linéaires**

On cherche à construire des codes polynomiaux de redondance minimale pour une dimension supérieure à 2, pouvant détecter 2 erreurs indépendantes et des erreurs indépendantes en nombre impair

1. Expliquez comment construire ces codes et exprimer pour chacun d ?eux son polynôme générateur et sa distance
2. Choisir un de ces codes et indiquer, en les justifiant, ses propriétés :
   (a) Quelle longueur maximum d ?un paquet d ?erreur peut-il détecter ?
   (b) Est-ce un code de Hamming ?
   (c) Est-ce un code parfait ?
   (d) Pour quelle longueur minimum est-il cyclique ?

**Exercice 39 – Sur les codes cycliques polynomiaux**

On considère les codes polynomiaux suivants : $C_1(5,4)$ défini par le polynôme générateur : $g_1(x) = x + 1$ et $C_2(5,1)$ défini par le polynôme générateur : $g_2(x) = x^4 + x^3 + x^2 + x + 1$

Caractériser ces deux codes

1. $C_1$ est-il cyclique ? Pourquoi ?
2. $C_2$ est-il cyclique ? Pourquoi ?
3. Existe-t-il d ?autres codes cycliques de même longueur ? Lesquels ?
4. Pour chacun de ces 2 codes $C_1$ et $C_2$ :
   (a) construire les matrices génératrices et de contrôle pour des codes systématiques équivalents. Justifier leur construction
   (b) calculer leurs distances et capacités de détection et de correction

**Exercice 40 – Sur les codes cycliques linéaires**

Le code $C(7,4)$ engendré par le polynôme $g(x) = 1 + x^2 + x^3$

1. est-il équivalent à un code de Hamming ? Justifiez.
2. est-ce un code cyclique ? Justifiez.
3. On notera les polynômes et la matrice génératrice $G$ de $C$ avec les poids faibles en tête. On construira la matrice $G$? équivalente à $G$ diagonale-gauche.

**Exercice 41 – Sur les codes cycliques linéiaires**

On considère le code cyclique $C(7,4)$ engendré par le polynôme générateur $g(x) = x^3 + x^2 + 1$

1. Montrer que le polynôme générateur est bien celui d'un code cyclique.
2. Ce code peut-il détecter :
   (a) toutes les erreurs simples ?
   (b) toutes les erreurs doubles ?



   (c) toutes les erreurs de poids impair ? Justifier chacune des réponses. On pourra utiliser des théorèmes connus en les citant.
3. Soit $f(x) = x^3 + 1$ un bloc à coder
   (a) Quel est le mot du code $c(x)$ correspondant ?
   (b) Calculer $h(x)$ tel que : $g(x) \times h(x) \mod (x^7 + 1) = 0$
4. On appelle syndrome d'un polynôme $a(x)$ l'expression : $s(a(x)) = h(x) \times a(x) \mod (x^7 + 1)$. Calculer le syndrome de $c(x)$
5. Soit $e(x) = (x^5 + x^3)$ un polynôme d'erreur.
   (a) Calculer le polynôme $d(x)$ reçu pour $c(x)$ transmis avec l'erreur $e(x)$
   (b) Calculer le syndrome de $d(x)$
   (c) L'erreur $e(x)$ est elle détectable ? Justifier