Dimanche 14 août 2016

# Jeux de cartes et de chapeaux, transmission de données et codes correcteurs d'erreurs.

*Michel Waldschmidt*

**Professeur émérite,**
**Université Pierre et Marie Curie (Paris 6)**

http://www.math.jussieu.fr/~miw/

# Aspects mathématiques de la théorie des codes en France:

Les principales équipes de recherche sont regroupées dans le réseau C2 ''Théorie des codes et cryptographie " , qui fait partie du groupe de recherche (GDR) ''Informatique Mathématique". http://www.gdr-im.fr/

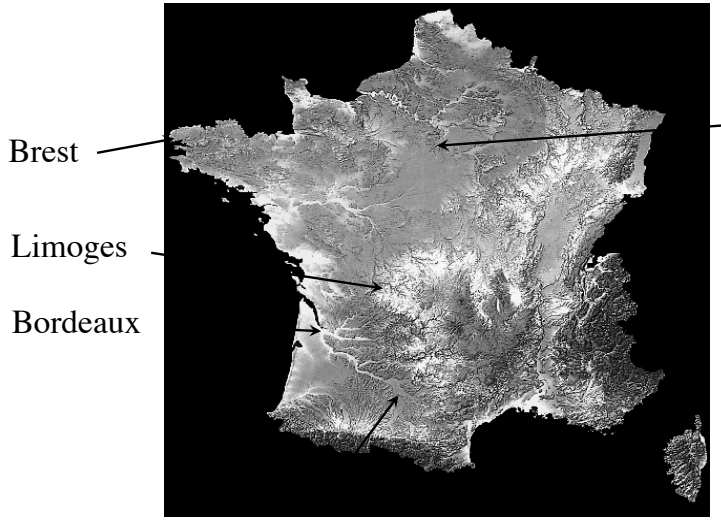


Recherche en théorie des codes

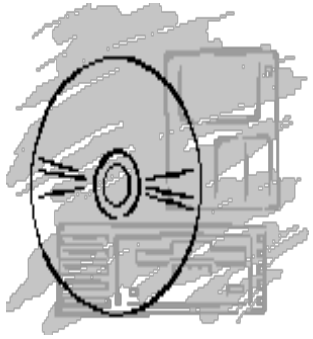
# Principaux centres:

INRIA Rocquencourt
Université de Bordeaux
ENST Télécom Bretagne
Université de Limoges
Université de Marseille
Université de Toulon
Université de Toulouse



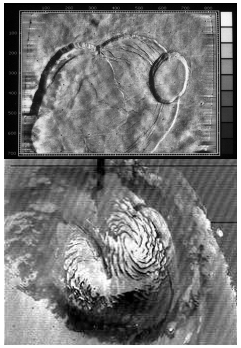


# *Codes correcteurs d'erreurs et transmission de données*



- Transmissions par satellites
- CD's & DVD's
- Téléphones cellulaires

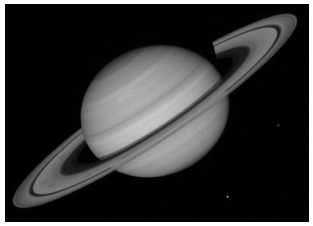




Mariner 2 (1971) et 9 (*1972*)

Le Mont Olympus sur la planète Mars

Le pôle nord de la planète Mars

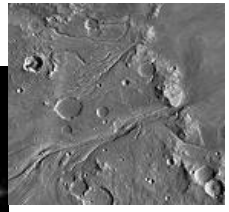


Voyager 1 et 2 (*1977*)

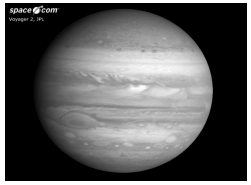
Trajet: Cap Canaveral, Jupiter, Saturn, Uranus, Neptune.



***Mariner 9 (1979)***

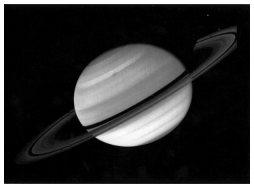


Photographies en noir et blanc de Mars



***Voyager (1979-81)***

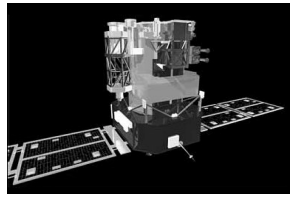
Jupiter Saturne





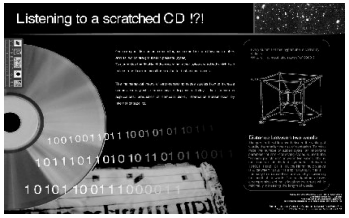
NASA : mission Pathfinder sur Mars (1997)

- *1998*: perte de contrôle du satellite Soho
  Récupération grâce à une double correction par un turbo code.



Les transmissions par radio sur ces engins spatiaux n'utilisent que quelques watts. Malgré l'importance du bruit qui vient perturber les messages, les transmissions sur des centaines de millions de km se font sans perte d'information.

Un CD de haute qualité a facilement plus de *500 000* erreurs!




- Le traitement du signal permet de corriger ces erreurs et d'annuler le bruit.
- Sans code correcteur d'erreurs, il n'y aurait ni CD ni DVD.



SONY

PHILIPS
sense and simplicity

*1* seconde de signal audio
= *1 411 200* bits

- 1980 : accord entre Sony et Philips pour une norme concernant les disques CD audio.
- 44 100 fois par seconde, 16 bits pour chacun des deux canaux stéréos



## *Codes et Mathématiques*


- Algèbre
  (mathématiques discrètes, algèbre linéaire,…)
- Géométrie
- Probabilités et statistiques



## *Corps finis et théorie des codes*

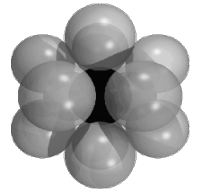
- Résolutions d'équations par radicaux: théorie des corps finis (Galois fields)
  ***Evariste Galois*** *(1811-1832)*
- Construction de polygones réguliers par la règle et le compas
- Théorie des groupes

## *Codes et Géométrie*

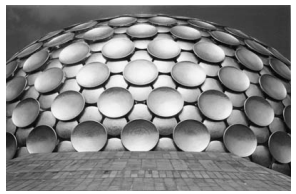
- *1949*: Marcel Golay (specialiste des radars): trouve deux codes remarquablement efficaces.
- Eruptions de Io (planète volcanique de Jupiter)
- *1963* John Leech utilise les idées de Golay's pour étudier les empilements de sphères en dimension *24* - *classification des groupes finis simples*.
- *1971*: il n'y a pas d'autre code *parfait* corrigeant plus d'une erreur que les deux trouvés par Golay.

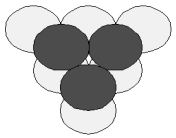


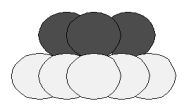
## Empilement de sphères



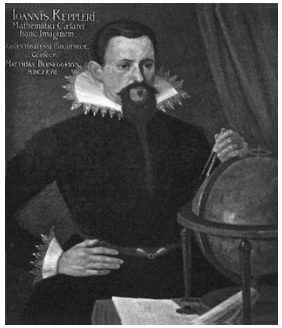
"kissing number" *12*












## Empilement de sphères

Problème de Kepler: densité maximale d'un pavage de l'espace par des sphères identiques

$$\pi / \sqrt{18} = 0.740\ 480\ 49\ldots$$

Conjecturé en *1611*.

Démontré en *1999* par *Thomas Hales*.

- Lien avec la cristallographie.



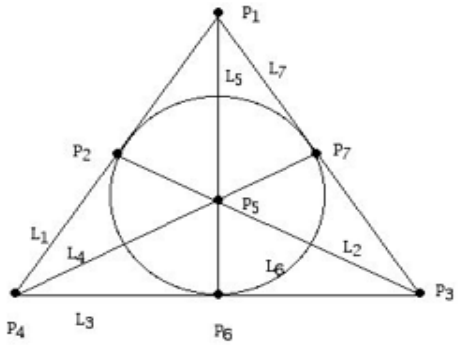
## Géométrie projective finie

Deux points déterminent une ligne (« droite »),
deux droites se coupent en un point.
Trois points sur chaque droite,
par chaque point passent trois droites.

Plan de Fano

Matrice d'incidence:

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



## *Quelques codes utiles*

- *1955:* Codes de convolution.
- *1959:* Bose Chaudhuri Hocquenghem (codes BCH).
- *1960*: Reed Solomon.
- *1970*: Goppa.
- *1981*: Géométrie algébrique



## Le problème des chapeaux

## *Le problème des chapeaux*

- Trois personnes, formant une équipe, ont chacune un chapeau sur la tête, blanc ou noir.
- Les couleurs sont choisies de façon aléatoire.
- Chacun voit la couleur des chapeaux sur la tête des deux autres, mais ne connaît pas la couleur de son propre chapeau.
- Chacun doit deviner la couleur de son chapeau en l'écrivant sur un papier: *blanc, noir, abstention*.



## *Règles du jeu*

- Les trois personnes forment une équipe, elles gagnent ou perdent ensemble.
- Elles ne communiquent pas, mais ont convenu d'une stratégie.
- L’équipe gagne si une au moins des trois personnes ne s’abstient pas, et si aucun de ceux ayant parié *blanc* ou *noir* ne s’est trompé.
- Quelle stratégie adopter pour optimiser les chances de gagner?

## *Stratégie*

- ***Une stratégie faible:*** chacun met une une réponse au hasard, *blanc* ou *noir*. Probabilité de gagner: $1/2^3 = 1/8$.
- ***Stratégie un peu meilleure:*** ils se mettent d'accord que deux d'entre eux s'abstiennent, le troisième donne une réponse au hasard. Probabilité de gagner: ***1/2***.
- Peut-on faire mieux?



## *La clé:*  *utiliser l'information disponible*

- ***Indication:***

  Augmenter ses chances en tenant compte de l'information disponible: chacun voit la couleur des chapeaux sur la tête des deux autres (mais pas sur la sienne).



## *Solution du problème des chapeaux*

- ***Stratégie***: si un membre de l'équipe voit deux chapeaux de couleurs différentes, il s'abstient.
- S'il voit deux chapeaux de la même couleur, il parie que le sien est de l'autre couleur.





Les deux personnes ayant un chapeau blanc voient un chapeau blanc et un noir, elles s'abstiennent.

La personne ayant un chapeau noir voit deux chapeaux blancs, elle écrit ***Noir***

***L'équipe gagne!***



Les deux personnes avec un chapeau noir voient un chapeau blanc et un noir, elles s'abstiennent.

La personne avec un chapeau blanc voit deux chapeaux noirs, elle écrit ***Blanc***

***L'équipe gagne!***





Chacun voit deux chapeaux blancs, tout le monde écrit ***Noir***

***L'équipe perd!***





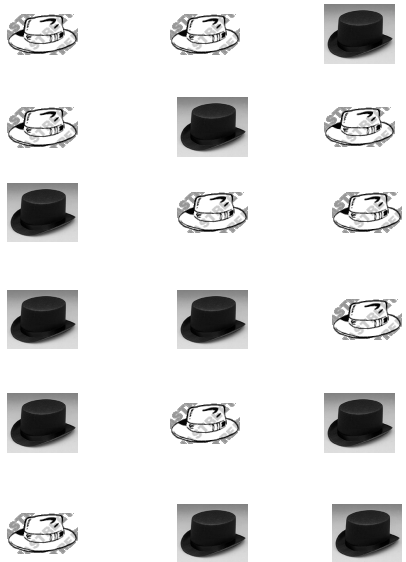
Chacun voit deux chapeau noirs, tout le monde écrit ***Blanc***

***L'équipe perd!***



***L'équipe gagne:***

Deux blancs et un noir
ou
deux noirs et un blanc



***L'équipe perd:***

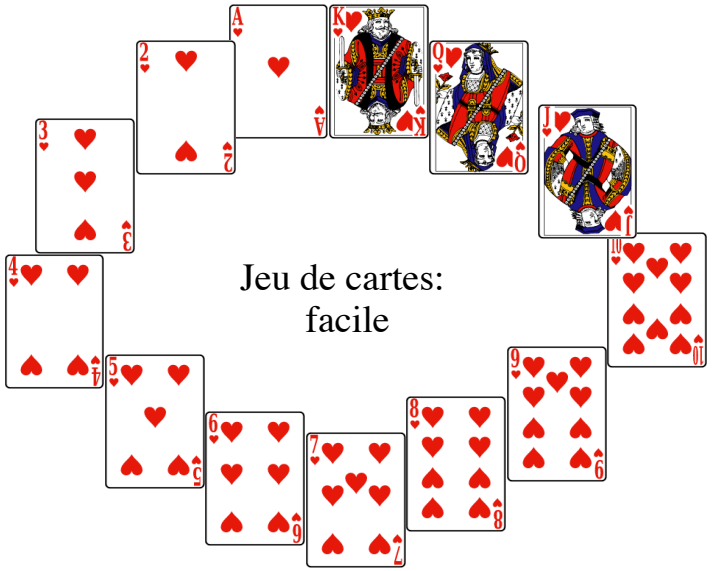


Trois blancs
ou
trois noirs



Probabilité de gagner : ***75%***







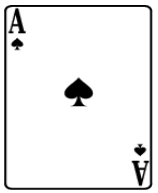

# Je sais quelle carte vous avez choisie

- Parmi une collection de cartes, vous en choisissez une sans me dire laquelle.
- Je vous pose des questions auxquelles vous répondez par oui ou non.
- Je peux déduire quelle carte vous avez choisie.
- Selon le nombre de cartes, combien de questions sont nécessaires? Et quelles questions suffisent?



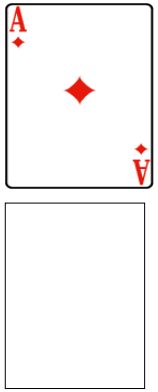
# *2* cartes

- Vous choisissez une des deux cartes
- Il me suffit d'une question à laquelle vous répondez oui ou non pour la connaître.

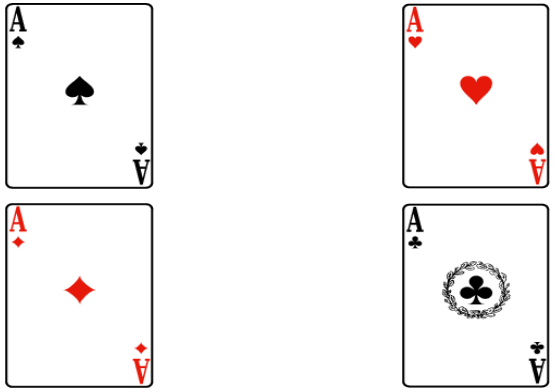
## *2* cartes: une question suffit

- Question: est-ce celle-ci?



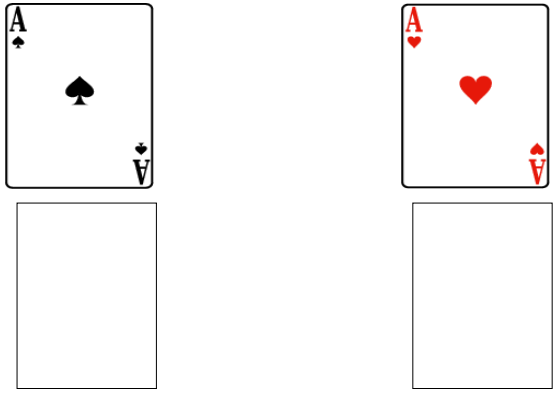


## *4* cartes





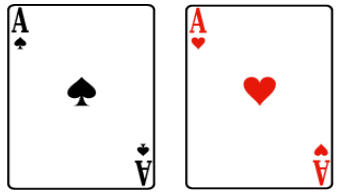
## Première question: est-ce une de ces deux cartes?



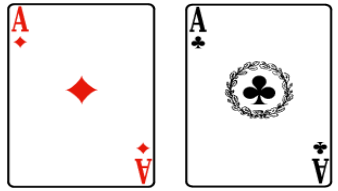


## Une fois connue la réponse à la première question on est ramené au problème précédent

- Si la première réponse est oui, il reste à trouver laquelle est la bonne parmi ces deux-ci



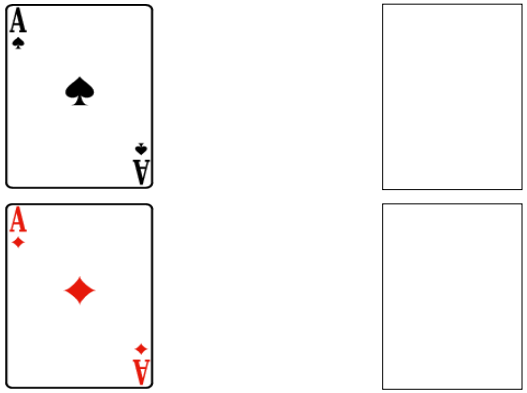
- Si la première réponse est non, il reste à trouver laquelle est la bonne parmi ces deux-là



## Deuxième question

(indépendante de la première réponse):

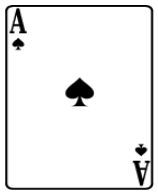
est-ce une de ces deux cartes?



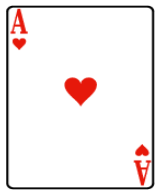


## *4* cartes: *2* questions suffisent

O O



O N

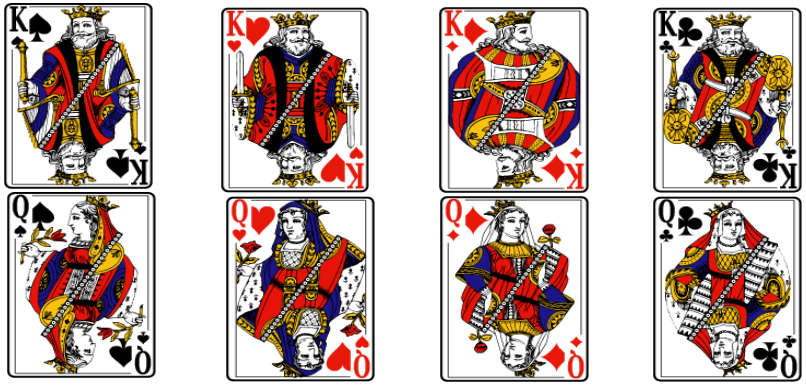


N O

N N



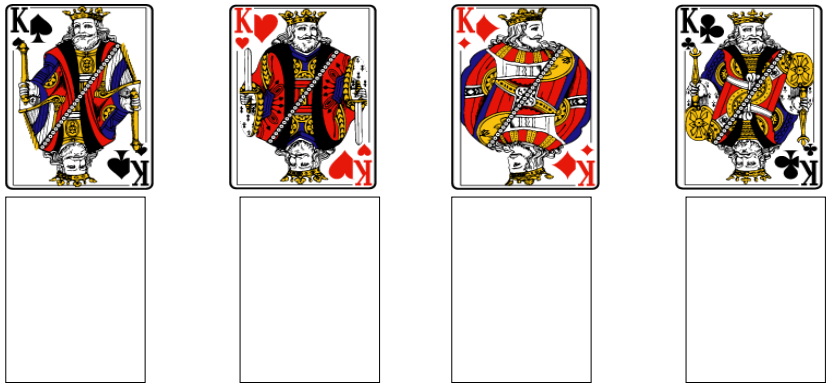
## *8* Cartes



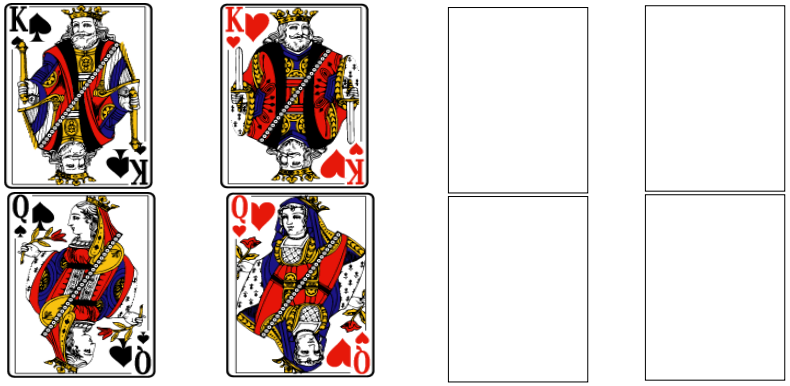


## Première question:

## est-ce une de ces quatre cartes?

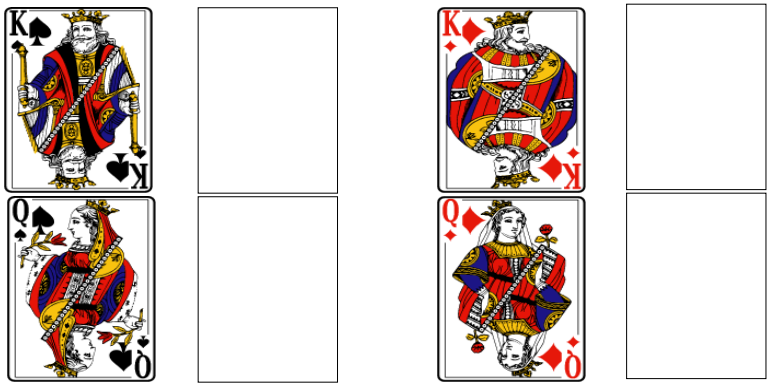


## Deuxième question:
## est-ce une de ces quatre cartes?





## Troisième question:
## est-ce une de ces quatre cartes?





陰 [阴]

陽 [阳]

## *8* cartes: *3* questions

OOO OON ONO ONN

NOO NON NNO NNN



## Oui / Non

- *0 / 1*
- Yin — / Yang - -
- Vrai / Faux
- Gauche / Droite
- Blanc / Noir
- \+ / -
- Pile / Face

# *8* Cartes: *3* questions

OOO OON ONO ONN

NOO NON NNO NNN

Remplacer O par 0 et N par 1



# *3* questions, *8* solutions

*0 0 0* *0 0 1* *0 1 0* *0 1 1*

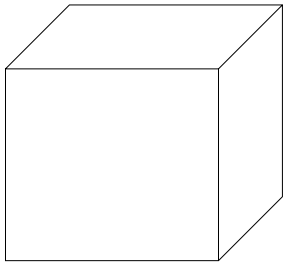
*0* *1* *2* *3*

*1 0 0* *1 0 1* *1 1 0* *1 1 1*

*4* *5* *6* *7*



$$8 = 2 \times 2 \times 2 = 2^3$$



On pourrait aussi disposer
les huit cartes aux sommets
d'un cube, plutôt que sur
deux lignes et quatre colonnes.



# Croissance exponentielle

*n* questions pour $2^n$ cartes

Une question de plus =
Deux fois plus de cartes

*Economie*:
Taux de croissance annuel de *4%* pendant *25* ans =
multiplier par 2,7

# Complexité

Un entier entre *0* et $2^n - 1$ est donné par son développement binaire qui compte *n* chiffres.

Notation binaire

$$m = a_{n-1}a_{n-2}\ldots a_1a_0$$

signifie

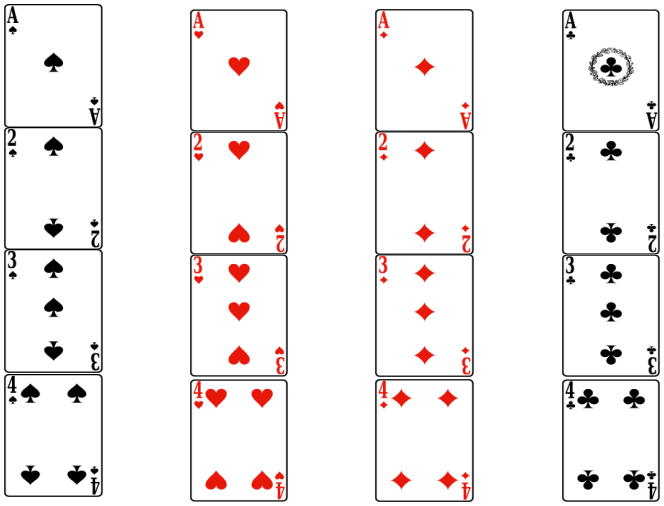
$$m = 2^{n-1}a_{n-1} + 2^{n-2}a_{n-2} + \ldots + 2a_1 + a_0.$$

La *complexité* de *m* est le nombre de chiffres :

$$n = 1 + [\log_2 m] \text{ si } a_{n-1} \neq 0.$$



# *16* cartes *4* questions



# Numéroter les *16* cartes

| | | | |
|---|---|---|---|
| *0* | *1* | *2* | *3* |
| *4* | *5* | *6* | *7* |
| *8* | *9* | *10* | *11* |
| *12* | *13* | *14* | *15* |



# Représentation binaire:

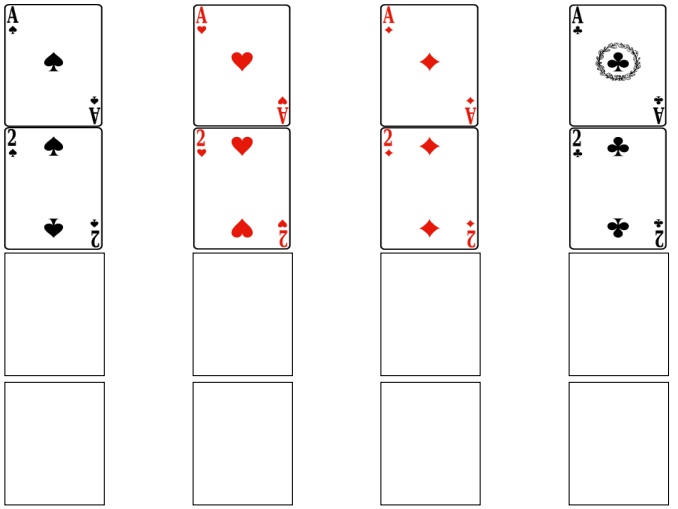
| | | | |
|---|---|---|---|
| *0 0 0 0* | *0 0 0 1* | *0 0 1 0* | *0 0 1 1* |
| *0 1 0 0* | *0 1 0 1* | *0 1 1 0* | *0 1 1 1* |
| *1 0 0 0* | *1 0 0 1* | *1 0 1 0* | *1 0 1 1* |
| *1 1 0 0* | *1 1 0 1* | *1 1 1 0* | *1 1 1 1* |

Poser les questions de telle sorte que les réponses soient:

| | | | |
|---|---|---|---|
| O O O O | O O O N | O O N O | O O N N |
| O N O O | O N O N | O N N O | O N N N |
| N O O O | N O O N | N O N O | N O N N |
| N N O O | N N O N | N N N O | N N N N |

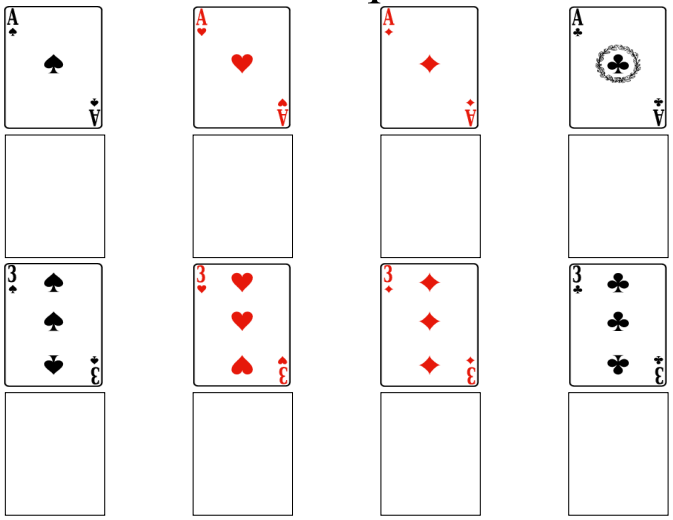


Première question:



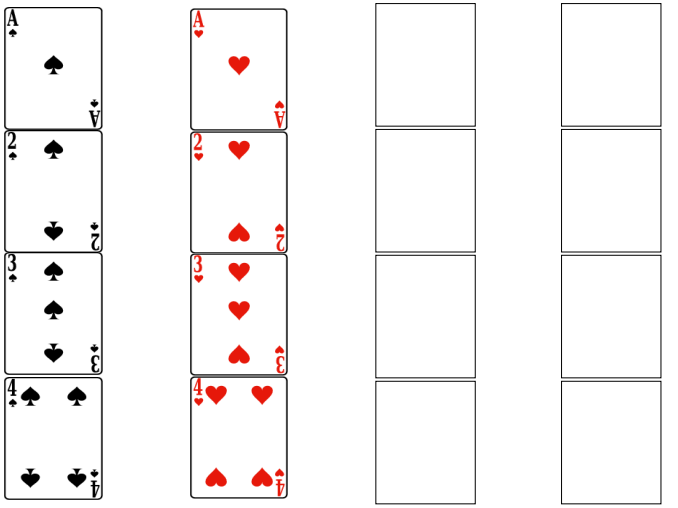


Deuxième question:

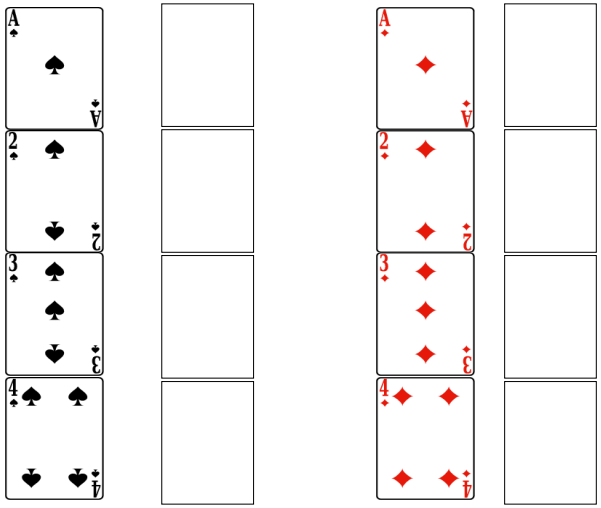




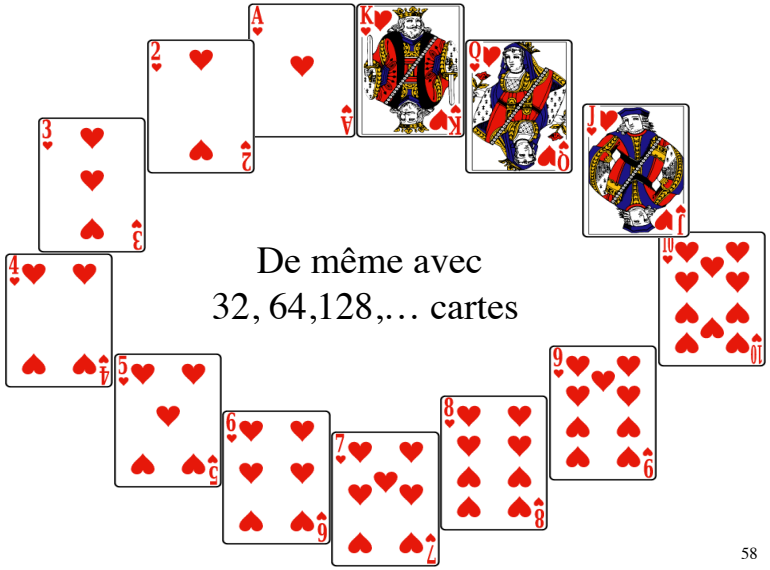
Troisième question:



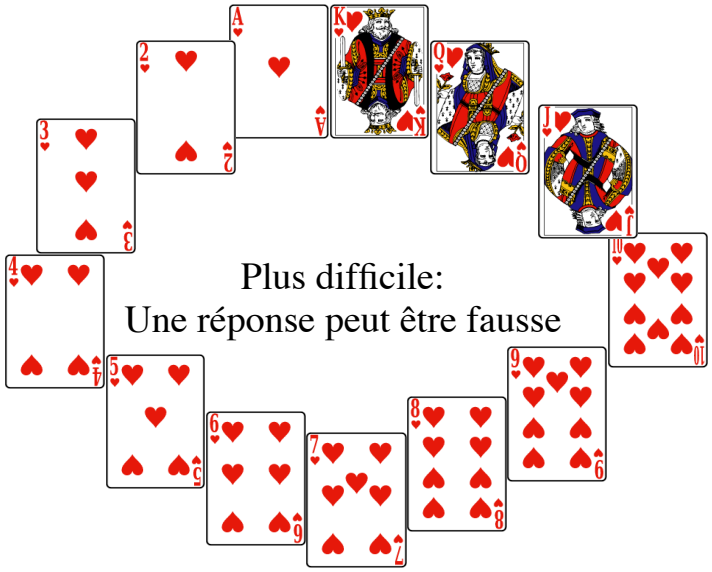
# Quatrième question:












# Une réponse peut être fausse

- On considère le même jeu, mais vous avez le droit de me donner une réponse fausse.
- Combien de questions sont nécessaires pour que je puisse dire s'il y a une réponse incorrecte? Et si toutes les réponses sont justes, je veux savoir quelle carte vous avez choisie.

## Détecter une erreur

- Il me suffit de poser une question de plus, cela me permet de détecter si une de vos réponse n'est pas compatible avec les autres.
- Et si toutes les réponses sont correctes, alors je sais quelle carte vous avez sélectionnée.



## Détecter une erreur avec *2* cartes

- S'il y a seulement deux cartes, il me suffit de répéter deux fois la même question.
- Si vos deux réponses sont identiques, alors elles sont correctes et je sais quelle carte vous avez sélectionnée.
- Si vos deux réponses ne sont pas identiques, je sais que l'une est correcte et pas l'autre, mais je n'en sais pas plus.



## Principe de la théorie des codes

Seuls certains mots sont autorisés *(**code** = dictionnaire des mots autorisés)*.

Les lettres « utiles » (*bits de données*) contiennent l'information, les autres (*bits de contrôle*) permettent de détecter (et parfois de corriger) des erreurs.



## Détecter une erreur en envoyant deux fois le même message

Envoyer chaque bit deux fois

*2* mots dans le code sur un nombre total de $4=2^2$ mots possibles

(*1* bit de données, *1* bit de contrôle)

Mots du code (longueur deux)

*0 0*

et

*1 1*

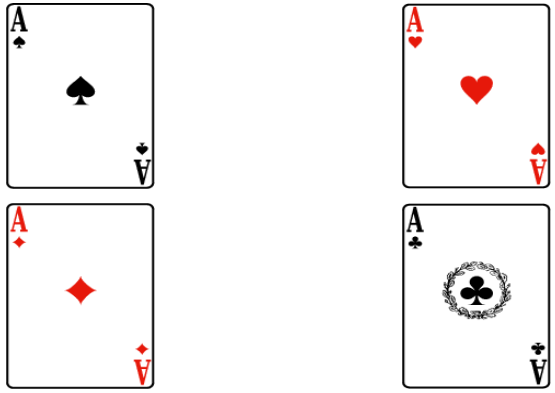
**Taux: *1/2***

Principe des codes détecteurs d'une erreur

*Deux mots distincts dans le code*
*ont au moins deux lettres distinctes*

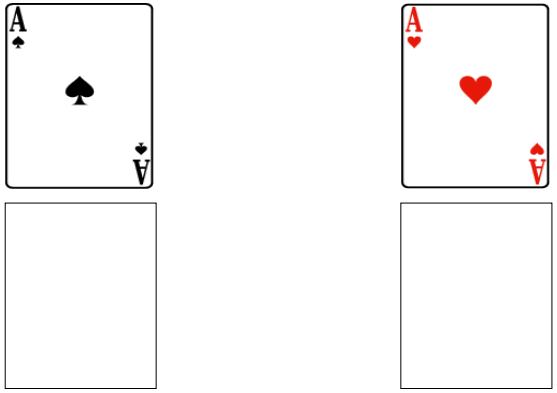


*4* cartes



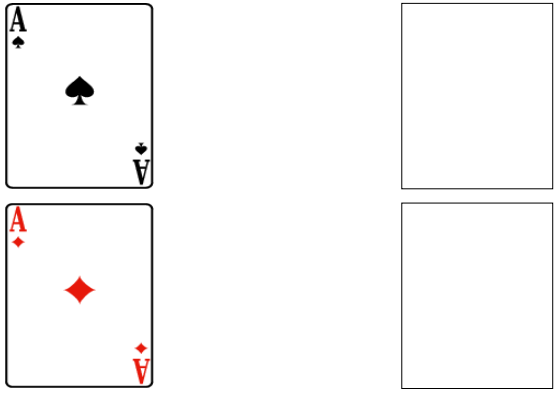


Première question:
est-ce l'une de ces deux cartes?

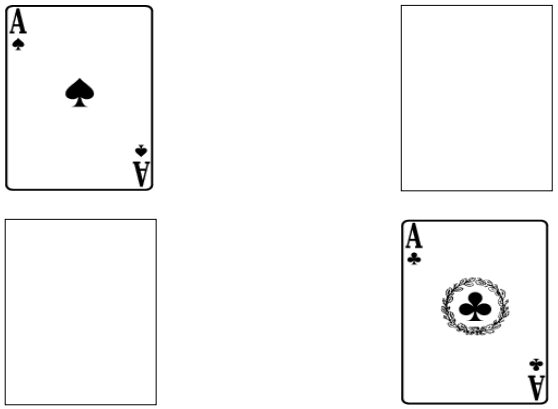




Deuxième question
est-ce l'une de ces deux cartes?



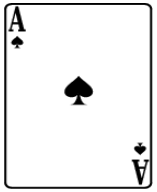
## Troisième question:
## est-ce l'une de ces deux cartes?



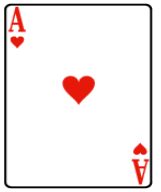


## *4* cartes: *3* questions

O O O



O N N



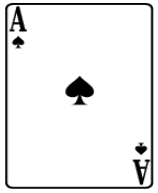
N O N

N N O

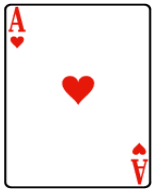


## *4* cartes: *3* questions

*0 0 0*



*0 1 1*



*1 0 1*

*1 1 0*



Triplets corrects de réponses

*0 0 0*  *0 1 1*  *1 0 1*  *1 1 0*

Triplets incorrects de réponses

*0 0 1*  *0 1 0*  *1 0 0*  *1 1 1*

Une modification dans un triplet correct
produit un triplet incorrect.

Dans un triplet correct, le nombre de *1* est pair,
dans un triplet incorrect, le nombre de *1* est impair,

# Addition Booléenne

- *0 + 0 = 0*
- *0 + 1 = 1*
- *1 + 0 = 1*
- *1 + 1 = 0*

- pair + pair = pair
- pair + impair = impair
- impair + pair = impair
- impair + impair = pair



# Bit de parité

- On introduit un bit supplémentaire qui est la somme Booléenne des précédents.
- Pour une réponse correcte la somme booléenne des bits est *0* (le nombre de *1* est pair).
- S'il y a exactement une erreur, le bit de parité la détecte: la somme Booléenne des bits est *1* au lieu d'être *0* (le nombre de *1* est impair).
- *Remarque:* le bit de parité permet de compléter un bit manquant.



# Bit de parité

- *L'International Standard Book Number* (ISBN) permet d'identifier des livres, le dernier des dix chiffres est un bit de parité.
- Le numéro de sécurité sociale comporte une clé qui permet de détecter une erreur.
- Les modems, les ordinateurs vérifient les données par un bit de parité.
- Les cartes de crédits utilisent des bits de parité.



# Détecter une erreur grâce au bit de parité

Mots du code (longueur *3*):

*0 0 0*

*0 1 1*

*1 0 1*

*1 1 0*

Bit de parité : *(*x y *z)* avec *z*=x+y.

*4* mots dans le code (il y a *8* mots de longueur *3*),

*2* bits de données, *1* bit de contrôle.

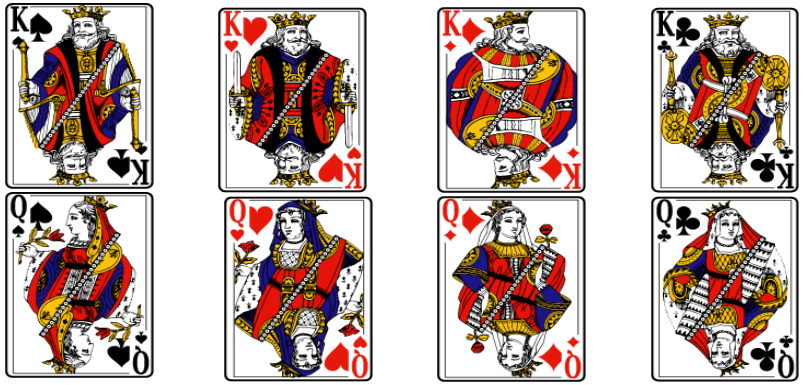
**Taux: *2/3***

| **Mots du code** | **Mots hors du code** |
|---|---|
| *0 0 0* | *0 0 1* |
| *0 1 1* | *0 1 0* |
| *1 0 1* | *1 0 0* |
| *1 1 0* | *1 1 1* |

*Deux mots du code distincts*
*ont au moins deux lettres distinctes.*



## *8* Cartes





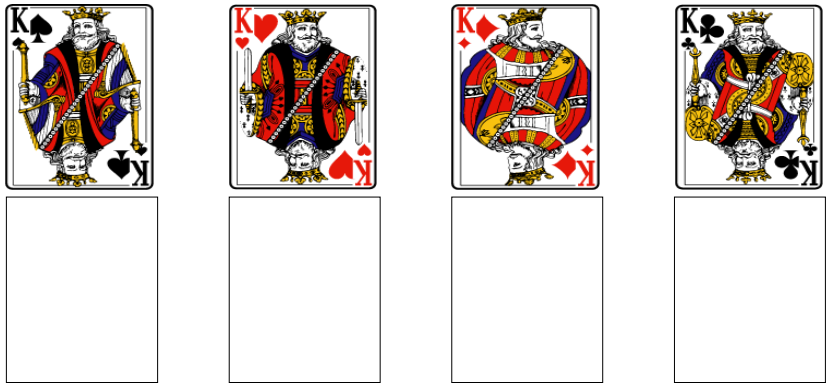
## *4* questions pour *8* cartes

On pose les *3* questions précédentes
plus la question du bit de parité
(le nombre de N doit être pair).

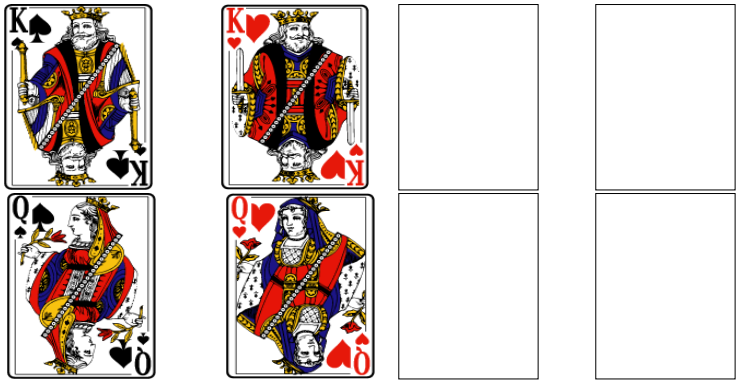
| | | | |
|---|---|---|---|
| *000**0*** | *001**1*** | *010**1*** | *011**0*** |
| OOO**O** | OON**N** | ONO**N** | ONN**O** |
| *100**1*** | *101**0*** | *110**0*** | *111**1*** |
| NOO**N** | NON**O** | NNO**O** | NNN**N** |



## Première question:
## Est-ce l'une de ces cartes?

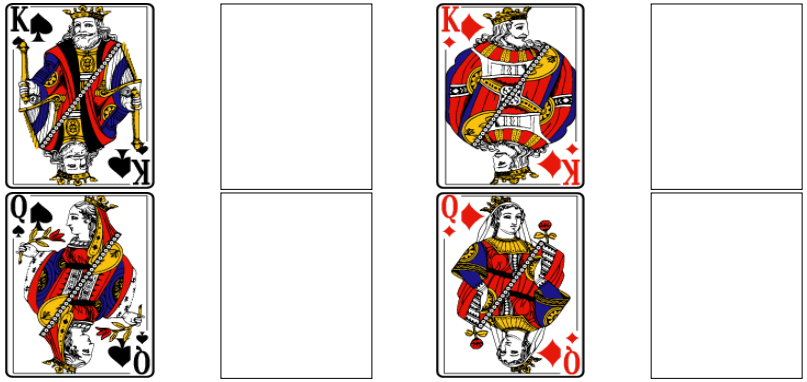


Deuxième question:
Est-ce l'une de ces cartes?



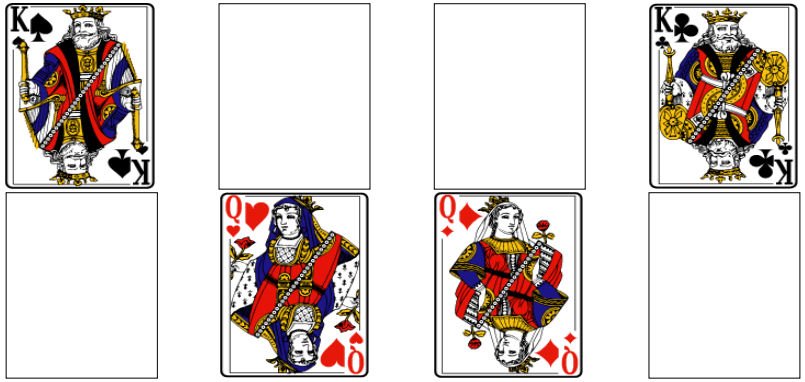
81

Troisième question:
Est-ce l'une de ces cartes?



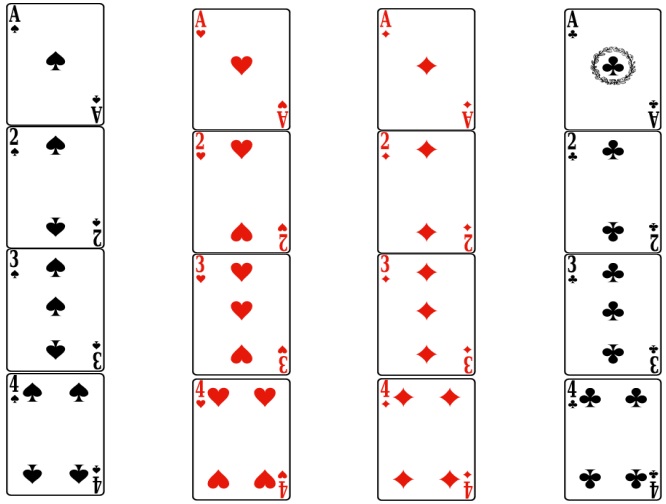
82

Quatrième question:
Est-ce l'une de ces cartes?



83

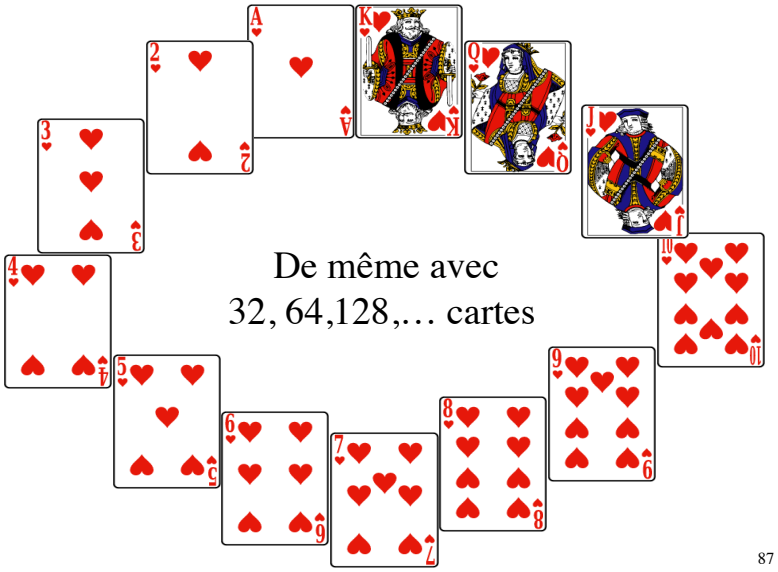
*16* cartes, au plus une erreur:
*5* questions pour la détecter



Poser les *5* questions
de telle sorte que les réponses soient:

| | | | |
|---|---|---|---|
| OOOOO | OOONN | OONON | OONNO |
| ONOON | ONONO | ONNOO | ONNNN |
| NOOON | NOONO | NONOO | NONNN |
| NNOOO | NNONN | NNNON | NNNNO |

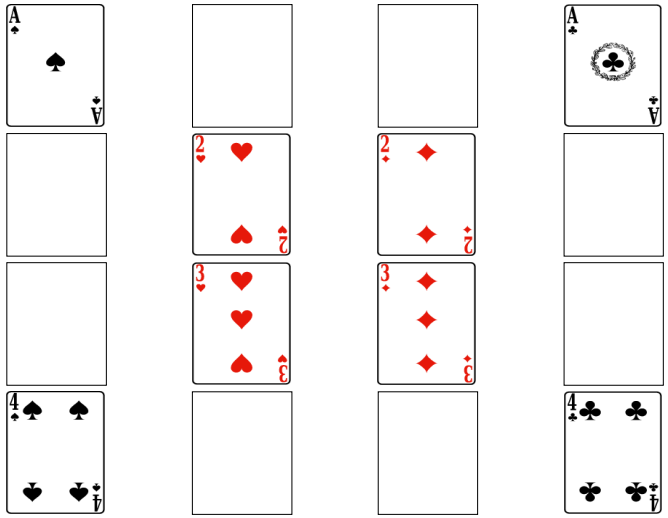




De même avec
32, 64,128,… cartes



*Cinquième question:*



## Corriger une erreur

- De nouveau je vous pose des questions auxquelles vous répondez oui ou non, une fois de plus vous pouvez me donner une réponse fausse, pas plus. Mais maintenant je veux pouvoir dire quelle carte vous avez choisie - en plus, je saurai si vous avez donné une réponse erronée, et dans ce cas je saurai laquelle c'est.

## Avec *2* cartes

- Je répète la même question trois fois.
- La réponse la plus fréquente est la bonne: *on vote avec la majorité*.
- *2* cartes, *3* questions, corrige *1* erreur.
- Réponses justes: *000* et *111*



# Corriger une erreur en répétant trois fois

- On envoie chaque bit trois fois

Mots du code (longueur trois)

*0 0 0*

*1 1 1*

*2* mots dans le code sur *8* possibles (*1* bit de données, *2* bits de contrôle)

**Taux: *1/3***



- Corriger *0 0 1* en *0 0 0*
- Corriger *0 1 0* en *0 0 0*
- Corriger *1 0 0* en *0 0 0*

et

- Corriger *1 1 0* en *1 1 1*
- Corriger *1 0 1* en *1 1 1*
- Corriger *0 1 1* en *1 1 1*



## Principe des codes corrigeant une erreur

*Deux mots distincts dans le code ont au moins trois lettres différentes*

# Distance de Hamming entre deux mots:

= nombre de bits où les deux mots diffèrent

***Exemples***

*(0,0,1)* et *(0,0,0)* sont à distance *1*

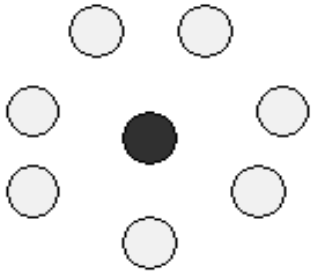
*(1,0,1)* et *(1,1,0)* sont à distance *2*

*(0,0,1)* et *(1,1,0)* sont à distance *3*

*Richard W. Hamming (1915-1998)*



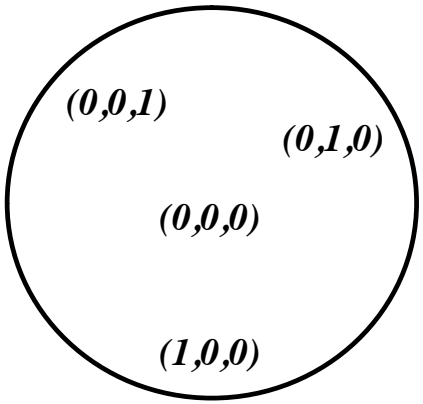
# Distance de Hamming égale à 1



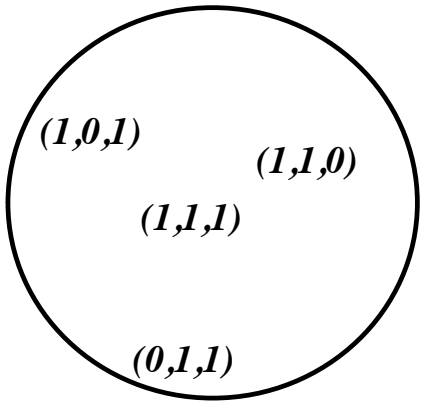
Mots obtenus en changeant un bit



Deux ou trois *0*




Deux ou trois *1*






# *Le code (0 0 0) (1 1 1)*

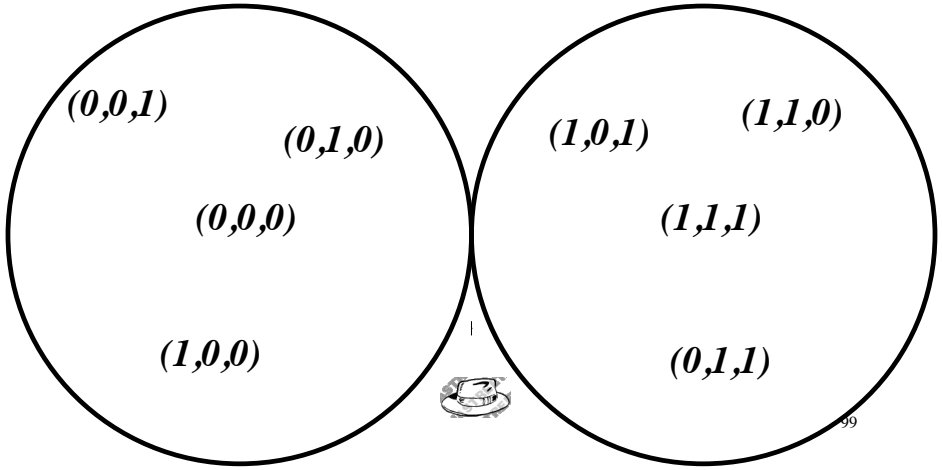
- L'ensemble des mots de longueur *3* (il y en a *8*) se répartit dans deux sphères de Hamming de rayon *1*.
- Les centres sont *(0,0,0)* et *(1,1,1)*
- Chacune des deux sphères est constituée des éléments à distance au plus *1* du centre.

# Retour au problème des chapeaux

Si un joueur voit deux ***0,***
Il sait que le centre
de la sphère est ***(0,0,0)***

Chaque joueur ne connaît que deux bits

Si un joueur voit deux ***1,***
il sait que le centre
de la sphère est ***(1,1,1)***



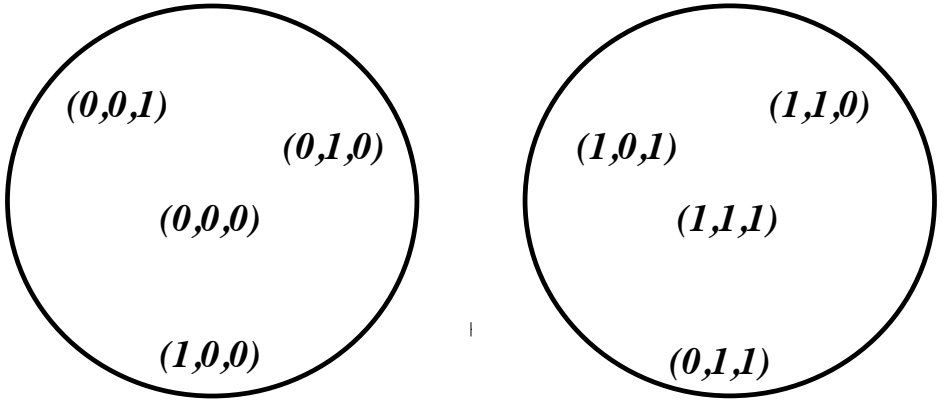



# Lien avec la théorie des codes

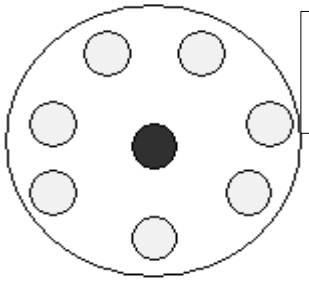
- On remplace blanc par *0* et noir par *1;*
  la répartition des couleurs des chapeaux devient un mot de longueur *3* sur l'alphabet *{0 , 1}*
- Considérer les sphères de rayon *1* et de centres *(0,0,0)* et *(1,1,1)*.
- L'équipe parie que la répartition des couleurs ne correspond pas à un des deux centres.



Si un joueur voit un ***0*** et un ***1,***
il ne sait pas quel est le centre
(mais il sait que l'équipe va gagner!)



# La sphère unité de Hamming



La sphère unité autour d'un mot

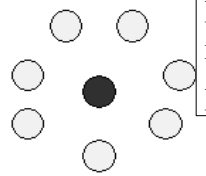
- La sphère unité de centre le mot bleu comporte les mots à distance au plus *1*



# Au plus une erreur

Mot envoyé

Le canal

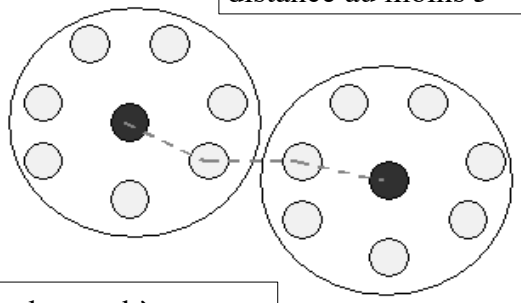


Mots qui peuvent être reçus avec au plus une erreur



# Mots à distance au moins *3*

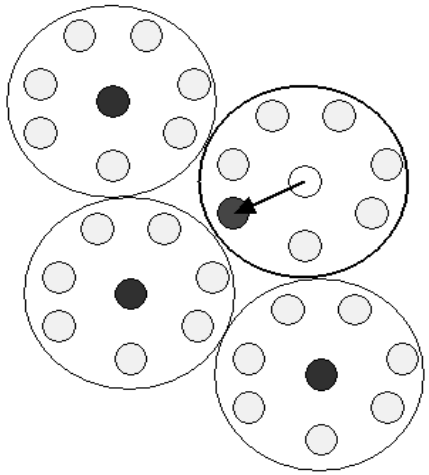
Ces mots sont à distance au moins *3*



Les deux sphères unités sont disjointes



# Décoder



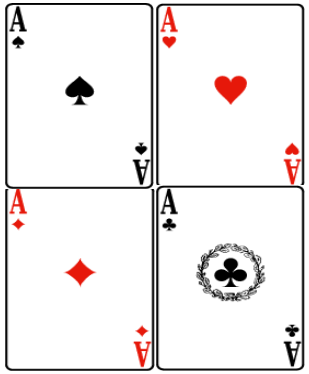
Le mot erroné reste dans la sphère de Hamming initiale, le centre est le mot du code

# Avec *4* cartes

Si je répète chacune des deux questions trois fois, il me faut *6* questions

Meilleure solution:
*5* questions suffisent

On répète chacune des questions précédentes seulement deux fois chacune, et on utilise le bit de parité.





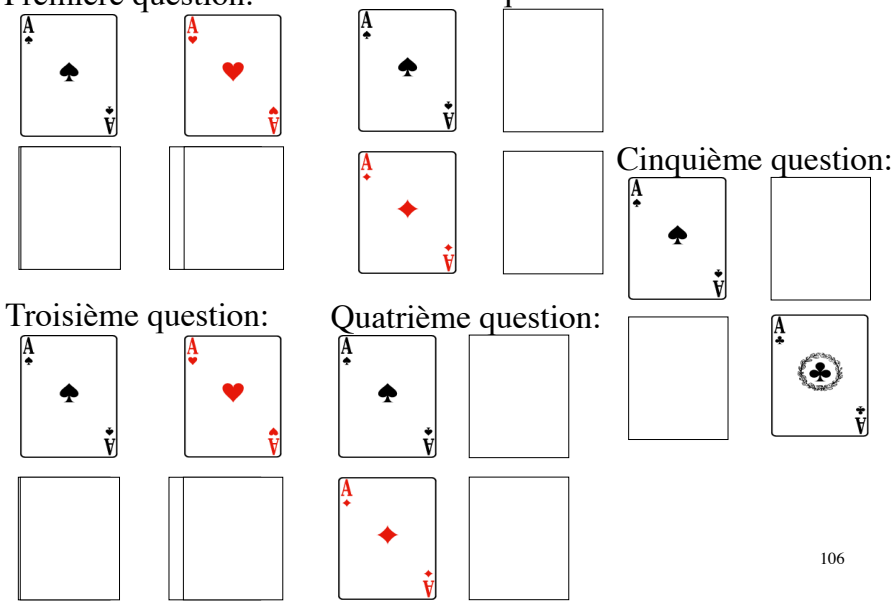
Première question:

Deuxième question:

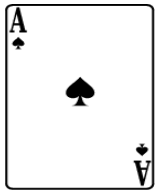
Troisième question:

Quatrième question:

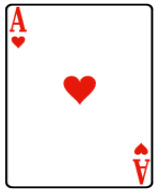
Cinquième question:







*4* cartes, *5* questions
corrige *1* erreur



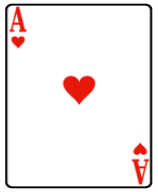
*4* réponses correctes: | *a* | *b* | *a* | *b* | *a*+*b* |

Au plus une erreur:
vous connaissez *a* ou *b*

Si vous connaissez ( *a* ou *b* ) et *a*+*b*
alors vous connaissez *a* et *b*

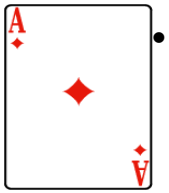
Longueur *5*

*2* bits de données,
*3* bits de contrôle



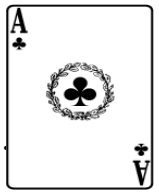
- *4* mots dans le code: *a*, *b*, *a*, *b*, *a*+*b*

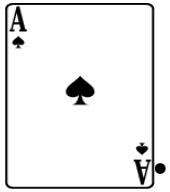
0 0 0 0 0
0 1 0 1 1
1 0 1 0 1
1 1 1 1 0



- Deux mots distincts du code sont à distance mutuelle au moins *3*

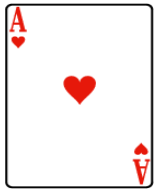
Taux : *2/5*





Longueur *5*

Nombre total de mots $2^5 = \mathbf{32}$



- *4* mots dans le code: $a, b, a, b, a+b$
- Chacun a *5* voisins
- Chacune des *4* sphères de rayon *1* a *6* éléments
- Il y a ***24*** réponses possibles comportant au plus *1* erreur
- ***8*** réponses ne sont pas possibles:

$$a, b, a+1, b+1, c$$

*(à distance ≥ 2 de chacun des mots du code)*

# Avec *8* cartes

Avec *8* cartes
et
*6* questions
on corrige
une erreur

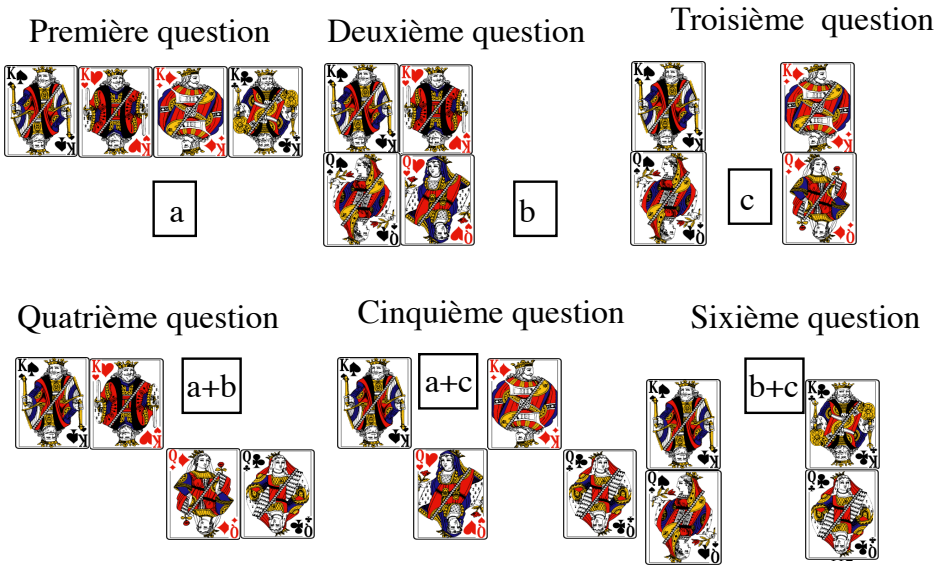


# *8* cartes, *6* questions, corrige *1* erreur

- On pose les trois questions qui fournissent la réponse s'il n'y a pas d'erreur, puis on utilise le bit de parité entre les questions (1,2), (1,3) et (2,3).
- Réponses correctes :

(a, b, c, a+b, a+c, b+c)

avec a, b, c remplacés par 0 ou 1



*8* cartes, *6* questions
corrige *1* erreur

- *8* réponses correctes: $a, b, c, a+b, a+c, b+c$
- avec $a, b, a+b$ on sait si $a$ et $b$ sont corrects
- Si on connaît $a$ et $b$, alors parmi $c, a+c, b+c$ il y a au plus une erreur, donc on connaît $c$

Longueur *6*

Nombre de mots $2^6$ ***=64***

- *8* mots dans le code: $a, b, c, a+b, a+c, b+c$
- Chacun a *6* voisins
- Chacune des *8* sphères de rayon *1* comporte *7* éléments
- Il y a ***56*** réponses possibles comportant au plus *1* erreur
- ***8*** réponses ne sont pas possibles:

$a, b, c, a+b+1, a+c+1, b+c+1$

*8* cartes, *6* questions
Corrige *1* erreur
*3* bits de données,
*3* bits de contrôle

- *8* mots dans le code: $a, b, c, a+b, a+c, b+c$

| | |
|---|---|
| 0 0 0 | 0 0 0 |
| 0 0 1 | 0 1 1 |
| 0 1 0 | 1 0 1 |
| 0 1 1 | 1 1 0 |
| 1 0 0 | 1 1 0 |
| 1 0 1 | 1 0 1 |
| 1 1 0 | 0 1 1 |
| 1 1 1 | 0 0 0 |

Deux mots distincts dans le code sont à distance au moins *3*

Taux : *1/2*.

# Nombre de questions

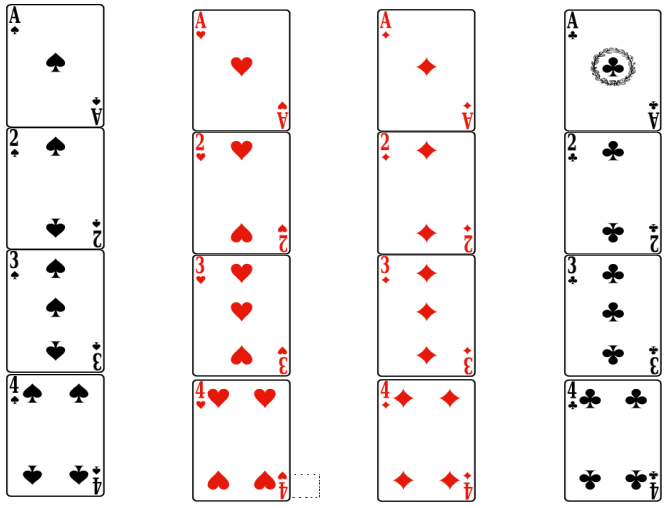
| | Pas d'erreur | Détecte *1* erreur | Corrige *1* erreur |
|---|---|---|---|
| *2* cartes | *1* | *2* | *3* |
| *4* cartes | *2* | *3* | *5* |
| *8* cartes | *3* | *4* | *6* |
| *16* cartes | *4* | *5* | *?* |

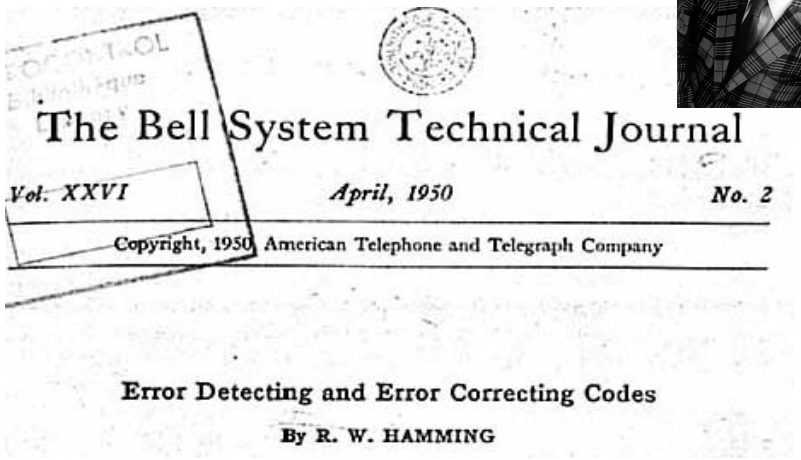
# Nombre de questions

| | Pas d'erreur | Détecte *1* erreur | Corrige *1* erreur |
|---|---|---|---|
| *2* cartes | *1* | *2* | *3* |
| *4* cartes | *2* | *3* | *5* |
| *8* cartes | *3* | *4* | *6* |
| *16* cartes | *4* | *5* | *7* |



# Avec *16* cartes, *7* questions permettent de corriger une erreur





The Bell System Technical Journal

Vol. XXVI April, 1950 No. 2

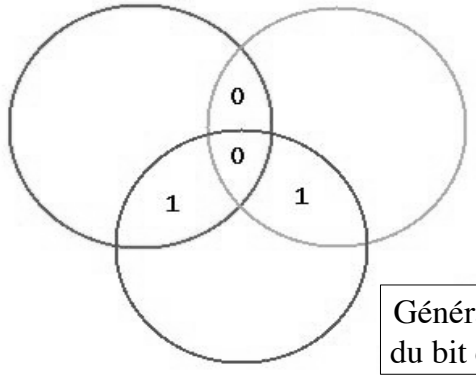


Error Detecting and Error Correcting Codes

By R. W. HAMMING



# Le code binaire de Hamming (1950)

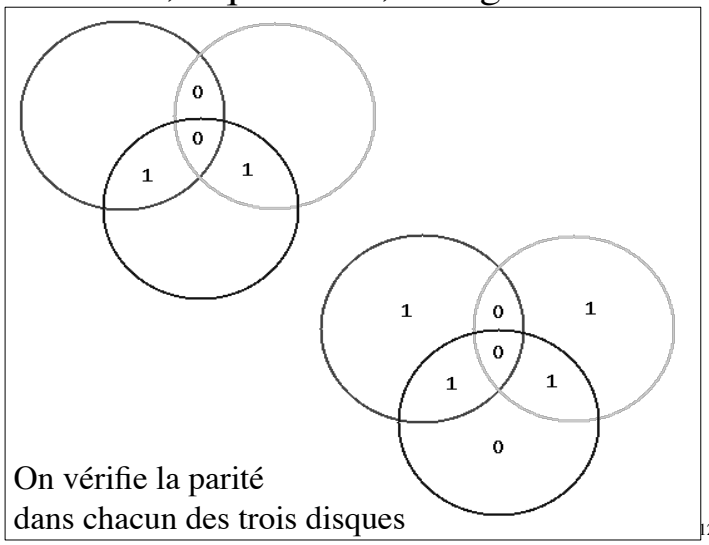



*4* questions précédentes,
*3* nouvelles,
corrige *1* erreur

On vérifie la parité
dans chacun
des trois disques.

Généralisation
du bit de parité

*16* cartes, *7* questions, corrige *1* erreur

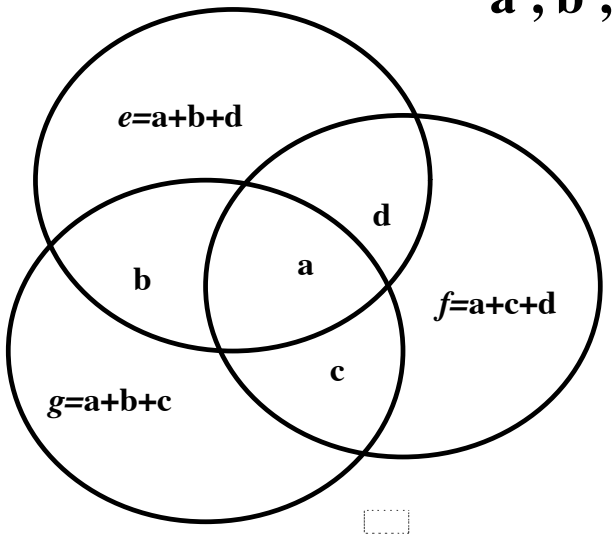



On vérifie la parité
dans chacun des trois disques



Calcul de *e* , *f* , *g* à partir de

**a , b , c , d**






# Code de Hamming

Mots de longueur *7*

Mots du code: (*16=2*$^4$ sur *128=2*$^7$ possibles)

*(*a, b, c, d, *e*, *f*, *g)*

avec

*e*=a+b+d

*f*=a+c+d

*g*=a+b+c

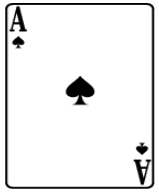
*4* bits de données, *3* bits de contrôle **Taux: 4/7**



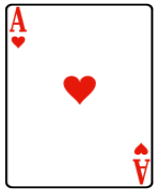
# *16* mots du code de longueur *7*

| | |
|---|---|
| 0 0 0 0 *0 0 0* | 1 0 0 0 *1 1 1* |
| 0 0 0 1 *1 1 0* | 1 0 0 1 *0 0 1* |
| 0 0 1 0 *0 1 1* | 1 0 1 0 *1 0 0* |
| 0 0 1 1 *1 0 1* | 1 0 1 1 *0 1 0* |
| 0 1 0 0 *1 0 1* | 1 1 0 0 *0 1 0* |
| 0 1 0 1 *0 1 1* | 1 1 0 1 *1 0 0* |
| 0 1 1 0 *1 1 0* | 1 1 1 0 *0 0 1* |
| 0 1 1 1 *0 0 0* | 1 1 1 1 *1 1 1* |

*Deux mots distincts dans le code*
*ont au moins trois lettres distinctes*



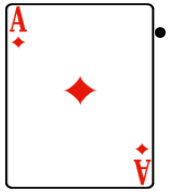
Mots de longueur *7*



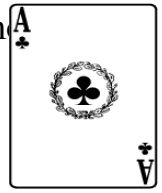
Nombre de mots: $2^7 = \mathbf{128}$

*Code de Hamming (1950)*:

- Il y a $16 = 2^4$ mots dans le code
- Chacun a *7* voisins
- Chacune des *16* sphères de rayon *1* a $8 = 2^3$ éléments et $16 \times 8 = 128$.



- Chacun des ***128*** mots est dans exactement une sphère:



empilement parfait

On numérote les cartes de *0* à *15* , on écrit les numéros en notation binaire

*0000, 0001, 0010, 0011*

*0100, 0101, 0110, 0111*

*1000, 1001, 1010, 1011*

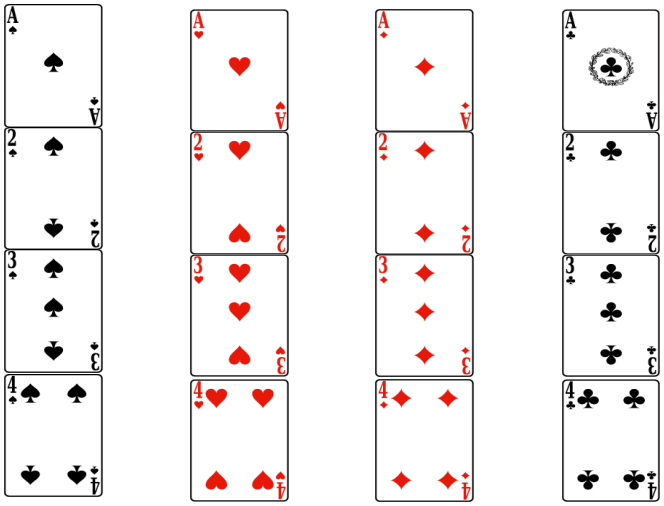
*1100, 1101, 1110, 1111*

grâce au code de Hamming on en déduit des mots de 7 bits.

On choisit les questions de telle sorte que Oui=0 et Non=1



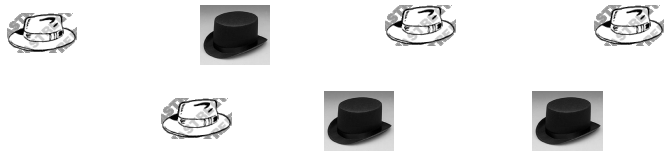
# *16* cartes , *7* questions corrige une erreur



*7* questions pour déterminer le nombre parmi *{0,1,2,…,15}* avec une erreur possible

- Le premier chiffre binaire est-il *0*?
- Le second chiffre binaire est-il *0*?
- Le troisième chiffre binaire est-il *0*?
- Le quatrième chiffre binaire est-il *0*?
- Est-il parmi *{1,2,4,7,9,10,12,15}*?
- Est-il parmi *{1,2,5,6,8,11,12,15}*?
- Est-il parmi *{1,3,4,6,8,10,13,15}*?

# Problème des chapeaux avec *7* personnes



Il y a maintenant *7* personnes au lieu de *3*,
quelle est la meilleure stratégie
et quelles sont les chances de gagner?

*Réponse:*
La meilleure stratégie
offre une probabilité de gagner de *7/8=87,5%*



# Jouer à la loterie

# *7* chapeaux

- L'équipe parie que la répartition des couleurs ne correspond pas à un des *16* éléments du code de Hamming
- L'équipe perd dans *16* cas (tout le monde se trompe)
- Elle gagne dans *128-16=112* cas (un seul a la bonne réponse, les *6* autres s'abstiennent)
- Probabilité de victoire : *112/128=7/8*



# Jouer à pile ou face

Lancer une pièce de monnaie *7* fois

Il y a $2^7=128$ suites possibles de résultats

Combien de paris faut-il faire pour être sûr que l'un au moins n'a pas plus d'un résultat faux?

# Pile ou face *7* fois de suite

- Chaque pari a tous les résultats justes une fois sur *128*.
- Il a exactement un prédiction fausse *7* fois: c'est soit la première, soit la seconde,… soit la septième.
- Il a donc au plus une prédiction fausse exactement *8* fois sur *128*.



# Lancer une pièce *7* fois

- Noter que *128 = 8 × 16*.
- On ne peut donc pas réussir avec moins que *16* paris.
- Le code de Hamming nous dit comment sélectionner les *16* paris de telle sorte que l'un d'eux aura au plus une prédiction erronée.



Principe des codes détectant *n* erreurs

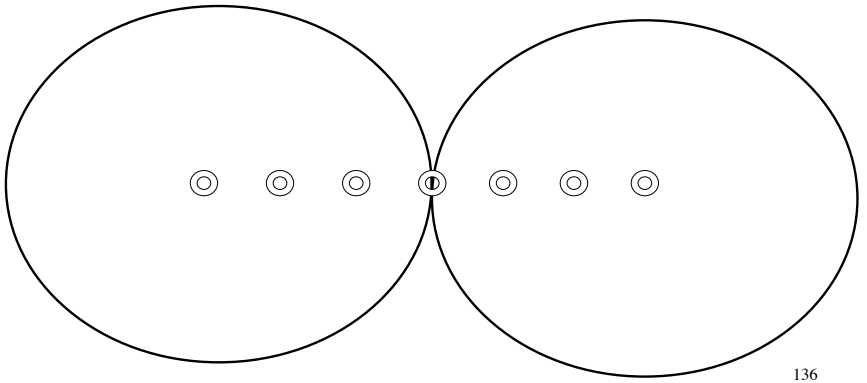
*Deux mots distincts du code ont*
*au moins n+1 lettres distinctes*

Principe des codes corrigeant *n* erreurs

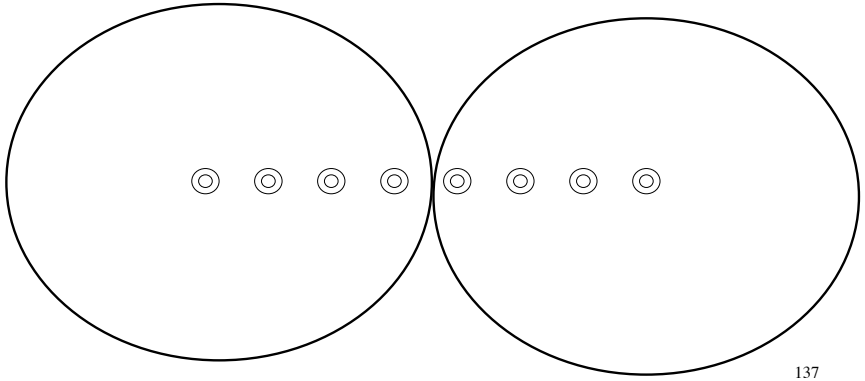
*Deux mots distincts du code ont*
*au moins 2n+1 lettres distinctes*



# Sphères de Hamming de rayon *3:* distance *6*, détecte *5* erreurs, corrige *2* erreurs



# Sphères de Hamming de rayon *3:* distance *7*, corrige *3* erreurs





# Code de Golay sur $\{0,1\}= \boldsymbol{F}_2$

Mots de longueur *23*, il y en a $2^{23}$ en tout
*12* bits de données, $2^{12}$ mots dans le code
*11* bits de contrôle,
distance *7*, corrige *3* erreurs
Chaque sphère de rayon *3* a

$$\binom{23}{0}+\binom{23}{1}+\binom{23}{2}+\binom{23}{3}$$
$$=1+23+253+1771=2048= 2^{11}$$

éléments, et $2^{12} \times 2^{11} = 2^{23}$ :

Empilement parfait



# Code de Golay sur $\{0,1,2\}= \boldsymbol{F}_3$

Mots de longueur *11*, il y en a $3^{11}$
*6* bits de données, *5* bits de contrôle,
distance *5*, corrige *2* erreurs
$3^6$ mots dans le code, chaque sphère
de rayon *2* a

$$\binom{11}{0}+ 2\binom{11}{1}+ 2^2\binom{11}{2}$$
$$=1+22+220=243= 3^5$$

éléments et $3^5 \times 3^6 = 3^{11}$ :

Empilement parfait



# SPORT TOTO: le plus ancien code correcteur d'erreurs

- Un match entre deux équipes ou deux joueurs peut donner trois résultats: ou bien le joueur *1* gagne, ou bien c'est le joueur *2*, ou bien il y a match nul (on écrit *0*).

- Un pari est gagnant s'il a au moins *3* prédictions correctes sur *4* matchs. Combien de tickets faut-il acheter pour être sûr que l'un d'eux est gagnant?

# *4* matchs, *3* prédictions justes

- Pour *4* matchs, il y a $3^4 = 81$ résultats possibles.
- Chaque pari pour *4* matchs est une suite de *4* symboles *{0, 1, 2}*. Chaque ticket a tout juste une seule fois, et exactement 3 prédictions correctes *8* fois.
- Donc chaque ticket est gagnant *9* fois sur *81*.
- Comme $9 \times 9 = 81$, il faut au moins *9* tickets pour être sûr de gagner.



*Journal de sports finlandais, 1932*

# *9 tickets*

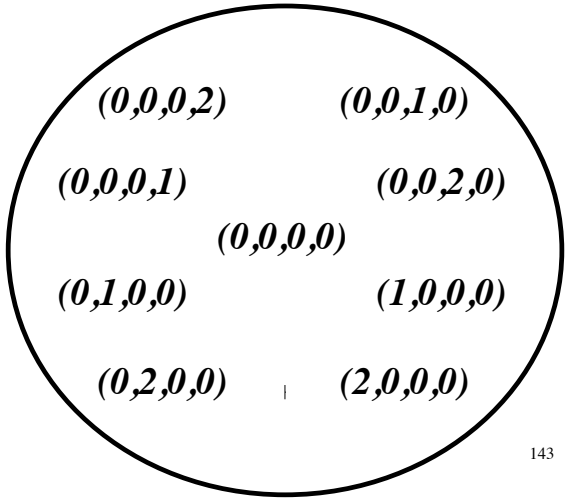
| | | |
|---|---|---|
| *0 0 0 0* | *1 0 1 2* | *2 0 2 1* |
| *0 1 1 1* | *1 1 2 0* | *2 1 0 2* |
| *0 2 2 2* | *1 2 0 1* | *2 2 1 0* |

Règle: *a, b, a+b, a+2b* modulo *3*

C'est un code correcteur d'erreur sur l'alphabet *{0, 1, 2}* avec comme taux *1/2*



# Empilement parfait de $\boldsymbol{F}_3^4$ avec *9* sphères de rayon *1*



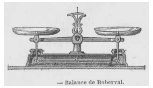



# Une fausse perle

- Parmi *9* perles d'apparence semblable, il y en a *8* vraies, identiques, ayant le même poids, et une fausse, qui est plus légère.
- Vous avez une balance permettant de comparer le poids de deux objets.
- En deux pesées vous pouvez déterminer la fausse perle.

# Pour trois perles: une pesée suffit

La fausse perle n'est pas pesée



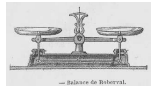
La fausse perle est à droite

La fausse perle est à gauche



# Pour *9* perles: on en met *3* à gauche et *3* à droite

La fausse perle n'est pas pesée



La fausse perle est à droite

La fausse perle est à gauche



# Chaque pesée permet de sélectionner le tiers de la collection où se trouve la fausse perle

- Première pesée: on prend *6* des *9* perles, on en met la moitié (*3)* de chaque côté de la balance.
- On détermine ainsi le groupe de *3* dans laquelle se trouve la fausse perle.
- Quand on a trois perles, une seule pesée suffit.



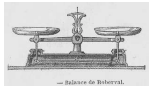
# Un protocole indépendant des résultats intermédiaires

- On numérote les *9* perles de *0* à *8* et on remplace ces numéros par leur écriture en base *3*.

| | | |
|---|---|---|
| *0 0* | *0 1* | *0 2* |
| *1 0* | *1 1* | *1 2* |
| *2 0* | *2 1* | *2 2* |

- Pour la première pesée, on met à gauche dans la balance les perles dont le numéro a pour premier chiffre *1* et à droite celles dont le numéro a pour premier chiffre *2*.

# Une pesée, un chiffre 0, 1 ou 2

La fausse perle n'est pas pesée  0

La fausse perle est à droite  1

La fausse perle est à gauche  2



# Résultat de deux pesées

- Chaque pesée produit trois résultats possibles: la balance est en équilibre *0*, ou bien elle est plus lourde à droite *1*, ou bien elle est plus lourde à gauche *2*.
- Les deux pesées produisent un nombre de deux chiffres en base *3* qui est le numéro de la fausse perle.



# *81* perles dont une fausse

- Pour *81* perles dont *80* vraies, identiques et une fausse qui est plus légère, quatre pesées permettent de déterminer la fausse perle.
- Pour $3^n$ perles dont une fausse, *n* pesées sont nécessaires et suffisantes.



# *Et si une des pesées donne un résultat erroné?*

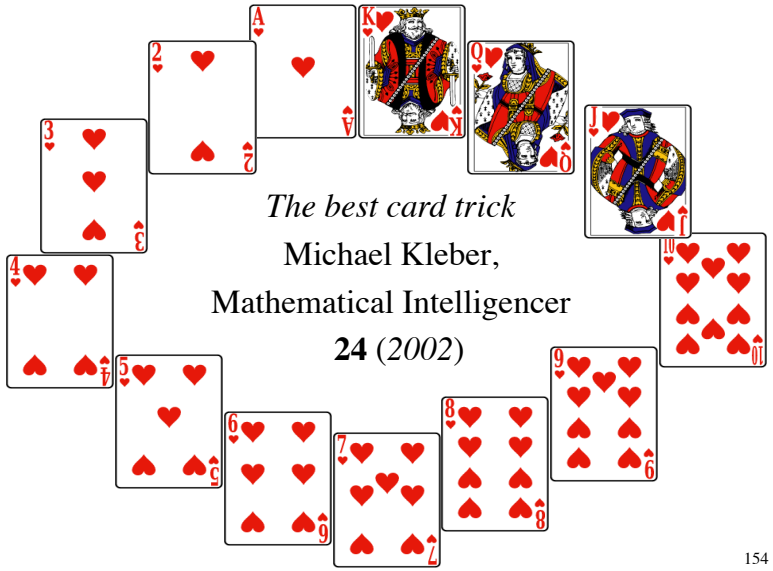
- Reprenons nos *9* perles. Si le résultat d'une des pesées risque d'être erroné, 4 pesées permettent quand même de déterminer la fausse perle.
- Pour cela on reprend le Sport Toto: on numérote nos *9* perles en recopiant les *9* tickets gagnants.

# Numérotation des 9 perles

| *a*, | *b*, | *a+b*, | *a+2b* | modulo *3* |
|---|---|---|---|---|
| *0* | *0* | *0* | *0* | |
| *0* | *1* | *1* | *1* | |
| *0* | *2* | *2* | *2* | |
| *1* | *0* | *1* | *2* | |
| *1* | *1* | *2* | *0* | |
| *1* | *2* | *0* | *1* | |
| *2* | *0* | *2* | *1* | |
| *2* | *1* | *0* | *2* | |
| *2* | *2* | *1* | *0* | |

Chacune des quatre pesées correspond à l'un des quatre chiffres, sur la balance on met à gauche les trois perles ayant le chiffre *1* et à droite les trois perles ayant le chiffre *2*





*The best card trick*
Michael Kleber,
Mathematical Intelligencer
**24** (*2002*)



# Règles du jeu

- Parmi *52* cartes à jouer, vous en selectionnez *5*, vous ne me les montrez pas, mais vous les donnez à mon assistant.
- Après les avoir regardées, il m'en donne *4*, l'une après l'autre. Il garde la cinquième sans me la montrer: seuls vous et lui la connaissez.
- Je suis alors capable de vous dire quelle est cette cinquième carte.



# Quelle information ai-je reçue?

- J'ai reçu *4* cartes, l'une après l'autre. Avec mon assistant nous nous sommes entendus préalablement sur l'ordre dans lequel il me les donnerait.
- Je peux ranger les *4* cartes que j'ai reçues de *24* façons différentes: j'ai *4* choix pour la première, une fois que je l'ai sélectionnée il me reste *3* choix pour la seconde, puis *2* pour la troisième, et je n'ai plus le choix pour la dernière.

$$24 = 4 \times 3 \times 2 \times 1$$

# *24* arrangements possibles pour *4* cartes

- Je peux donc convertir l'information que j'ai reçue en un nombre entre *1* et *24*.
- **Mais il y a 52 cartes!**
- J'en ai reçu *4*, il reste *48* possibilités pour la carte secrète.
- Avec un nombre entre *1* et *24*, je suis seulement à mi-chemin de la solution.
- Si nous convenions par exemple que le numéro de la carte est entre *1* et *24* quand il me les donne de la main droite et entre *25* et *48* s'il me les donne de la main gauche, ce serait tricher!



# Il n'y a que 4 couleurs!

Pique, Coeur, Carreau, Trèfle

♠ ♥ ♦ ♣

Mon assistant a reçu *5* cartes.

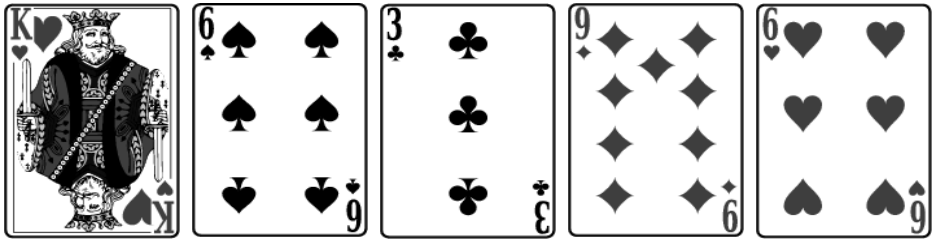


# Le principe des tiroirs

- S'il y a plus de pigeons que de trous, l'un au moins des trous héberge plusieurs pigeons.
- S'il y a plus de trous que de pigeons, l'un au moins des trous est vide.

Principe des tiroirs de Dirichlet
*box principle, pigeonhole principle, Schubfachprinzip*
1834.





Mon assistant a reçu *5* cartes, il y a *4* couleurs, donc au moins une des couleurs apparaît au moins deux fois.

Nous convenons que la couleur de la carte secrète sera la même que la couleur de la première carte qu'il me donne.

# Information que je reçois avec les trois autres cartes

- Il me reste à trouver quelle est la carte secrète parmi les *12* autres cartes de la même couleur que la première.
- Je reçois ensuite *3* cartes, il y a *6* ordres possibles, je peux convertir l'information reçue en un nombre entre *1* et *6*.



# Dernière étape

- Je dispose d'un nombre entre *1* et *6*, il y a *12* cartes possible, donc je suis encore à mi-chemin - mais j'ai progressé en réduisant le nombre de possibilités par un coefficient *4*, passant de *48* à *12*.
- Mon assistant a le choix au début, pour celle qu'il me donne en premier, entre deux cartes (au moins).