Formation à l'algorithmique
Activité télé-vision

Lycée Les 3 Sources ; Romain Janvier

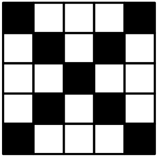
La télé-vision

- Se mettre par groupes de 3.

Émetteur	Transmetteur	Récepteur

La télé-vision

- Se mettre par groupes de 3.

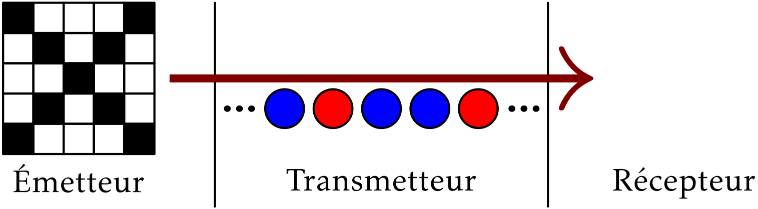


Émetteur | Transmetteur | Récepteur

- **L'émetteur** voit une image et doit la transmettre au **récepteur** en lui envoyant des jetons de couleurs, un par un.

La télé-vision

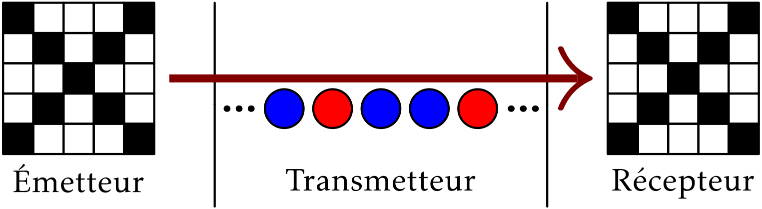
- Se mettre par groupes de 3.



- **L'émetteur** voit une image et doit la transmettre au **récepteur** en lui envoyant des jetons de couleurs, un par un.

La télé-vision

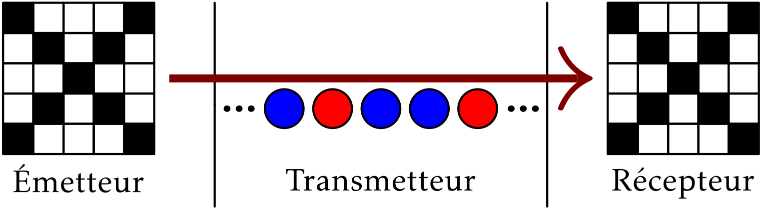
- Se mettre par groupes de 3.



- **L'émetteur** voit une image et doit la transmettre au **récepteur** en lui envoyant des jetons de couleurs, un par un.

La télé-vision

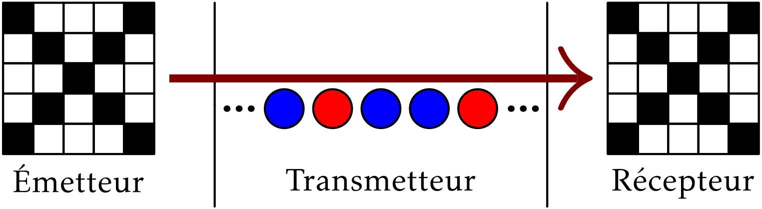
- Se mettre par groupes de 3.



- **L'émetteur** voit une image et doit la transmettre au **récepteur** en lui envoyant des jetons de couleurs, un par un.
- Il est interdit de parler ou de donner des indications en faisant des signes ou du bruit.

La télé-vision

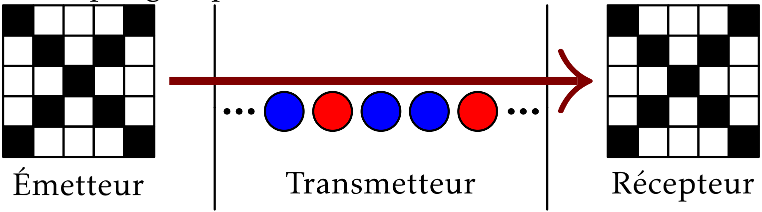
- Se mettre par groupes de 3.



- **L'émetteur** voit une image et doit la transmettre au **récepteur** en lui envoyant des jetons de couleurs, un par un.
- Il est interdit de parler ou de donner des indications en faisant des signes ou du bruit.
- Pour s'assurer que personne ne triche, **le transmetteur** reçoit les jetons de l'émetteur et les passe au récepteur.

La télé-vision

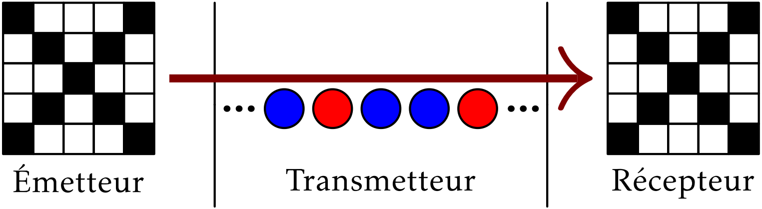
- Se mettre par groupes de 3.



- **L'émetteur** voit une image et doit la transmettre au **récepteur** en lui envoyant des jetons de couleurs, un par un.
- Il est interdit de parler ou de donner des indications en faisant des signes ou du bruit.
- Pour s'assurer que personne ne triche, **le transmetteur** reçoit les jetons de l'émetteur et les passe au récepteur.
- Vous devez vous mettre d'accord sur le **protocole de transmission** et ensuite demander une image à transmettre.

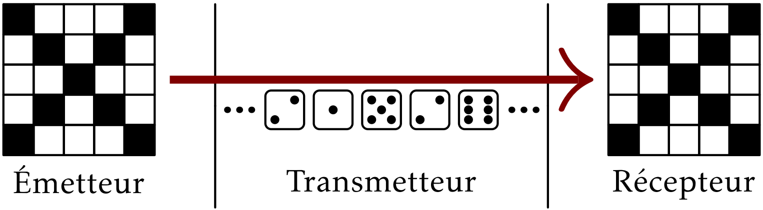
La télé-vision

- Se mettre par groupes de 3.



- **L'émetteur** voit une image et doit la transmettre au **récepteur** en lui envoyant des jetons de couleurs, un par un.
- Il est interdit de parler ou de donner des indications en faisant des signes ou du bruit.
- Pour s'assurer que personne ne triche, **le transmetteur** reçoit les jetons de l'émetteur et les passe au récepteur.
- Vous devez vous mettre d'accord sur le **protocole de transmission** et ensuite demander une image à transmettre.
- Changer les rôles et demander une 2e image à transmettre.

Version avancée



- Le problème est le même, mais à la place de jetons, vous avez des dés.
- Il faut trouver comment utiliser le moins de dés possible.

Erreur de transmission

- On reprend les jetons.
- Le transmetteur a le droit de changer la couleur d'un seul jeton pendant la transmission.

Erreur de transmission

- On reprend les jetons.
- Le transmetteur a le droit de changer la couleur d'un seul jeton pendant la transmission.
- Comment modifier le code pour détecter s'il y a eu une inversion ou pas ?

Erreur de transmission

- On reprend les jetons.
- Le transmetteur a le droit de changer la couleur d'un seul jeton pendant la transmission.
- Comment modifier le code pour détecter s'il y a eu une inversion ou pas ?
- Comment modifier le code pour corriger l'erreur s'il y en a une ?

Bilan de l'activité

- Avec les jetons :

Bilan de l'activité

- Avec les jetons :
  - Plusieurs types de parcours de l'image possibles.
  - Solution avec 25 jetons.
  - Tentatives de réduire le nombre de jeton en faisant des suppositions à priori (symétrie, lignes vides. . .)
  - Nombre de jetons minimum nécessaire dans le pire des cas ?

Bilan de l'activité

- Avec les jetons :
  - Plusieurs types de parcours de l'image possibles.
  - Solution avec 25 jetons.
  - Tentatives de réduire le nombre de jeton en faisant des suppositions à priori (symétrie, lignes vides. . .)
  - Nombre de jetons minimum nécessaire dans le pire des cas ?
- Avec les dés :
  - Utiliser 2 faces de chaque dés → 25 dés.
  - Coder des blocs avec un ou quelques dés.
  - Bien plus difficile.
  - Nombre minimal de dés ?

Bilan de l'activité

- Avec les jetons :
  - Plusieurs types de parcours de l'image possibles.
  - Solution avec 25 jetons.
  - Tentatives de réduire le nombre de jeton en faisant des suppositions à priori (symétrie, lignes vides. . .)
  - Nombre de jetons minimum nécessaire dans le pire des cas ?
- Avec les dés :
  - Utiliser 2 faces de chaque dés → 25 dés.
  - Coder des blocs avec un ou quelques dés.
  - Bien plus difficile.
  - Nombre minimal de dés ?
- Codes correcteurs :
  - . . .

Transmission des données numériques

- Omniprésence du numérique :

Transmission des données numériques

- Omniprésence du numérique :
  - Internet
  - Téléphones
  - TNT
  - Wifi
  - Bluetooth
  - USB
  - ...

Transmission des données numériques

- Omniprésence du numérique :
    - Internet
    - Téléphones
    - TNT
    - Wifi
    - Bluetooth
    - USB
    - ...
- Problématique majeure :
    - Limiter la taille des données.
    - Corriger un certain nombre d'erreurs de transmission.

Nombre minimal de jetons

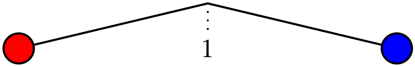
- Nombre d'images possibles :

Nombre minimal de jetons

- Nombre d'images possibles : $2^{25} = 33\,554\,432$.

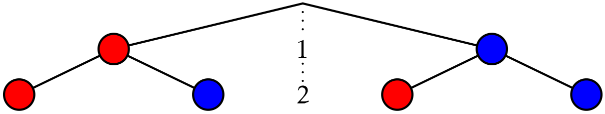
Nombre minimal de jetons

- Nombre d'images possibles : $2^{25} = 33\,554\,432$.
- À chaque pixel transmis, le nombre d'images possibles est divisé par 2.



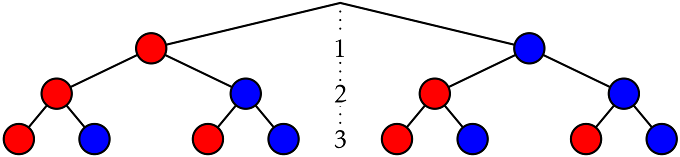
Nombre minimal de jetons

- Nombre d'images possibles : $2^{25} = 33\,554\,432$.
- À chaque pixel transmis, le nombre d'images possibles est divisé par 2.



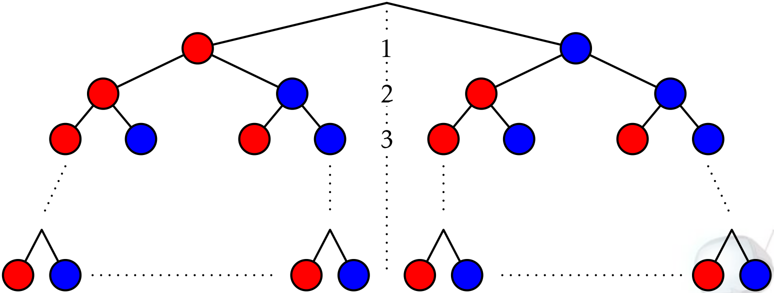
Nombre minimal de jetons

- Nombre d'images possibles : $2^{25} = 33\,554\,432$.
- À chaque pixel transmis, le nombre d'images possibles est divisé par 2.



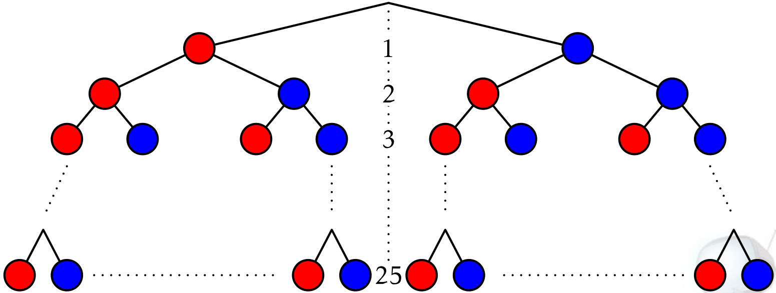
Nombre minimal de jetons

- Nombre d'images possibles : $2^{25} = 33\,554\,432$.
- À chaque pixel transmis, le nombre d'images possibles est divisé par 2.



Nombre minimal de jetons

- Nombre d'images possibles : $2^{25} = 33\,554\,432$.
- À chaque pixel transmis, le nombre d'images possibles est divisé par 2.



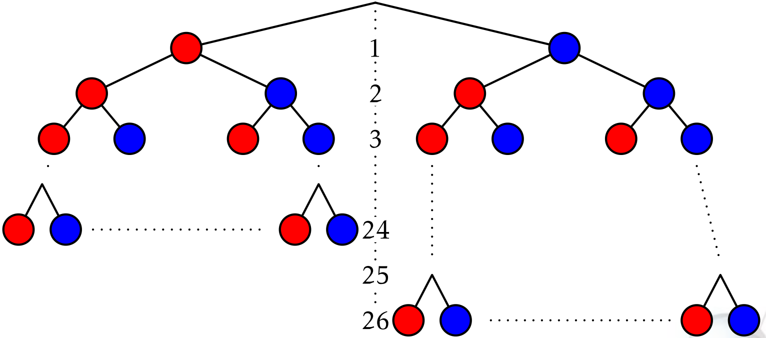
- Il faut 25 jetons pour pouvoir transmettre n'importe quelle image.

Code “déséquilibré”

- On peut utiliser moins de jetons pour certaines images, mais il en faut donc plus pour d'autres.

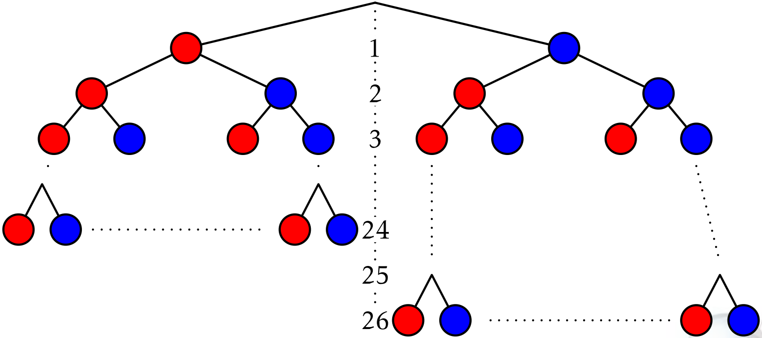
Code "déséquilibré"

- On peut utiliser moins de jetons pour certaines images, mais il en faut donc plus pour d'autres.



Code “déséquilibré”

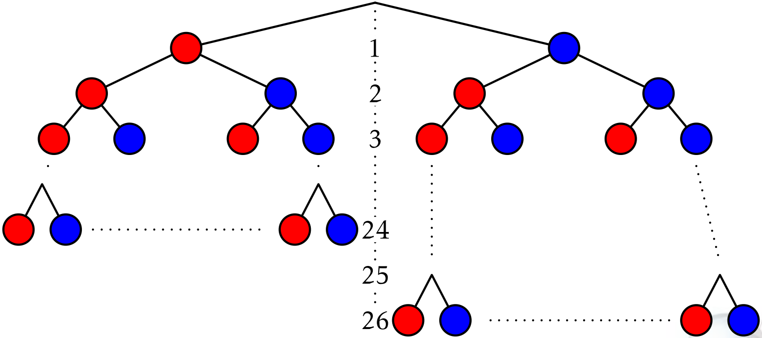
- On peut utiliser moins de jetons pour certaines images, mais il en faut donc plus pour d'autres.



- C'est le pire des cas, ou le cas moyen, qu'il faut étudier.

Code “déséquilibré”

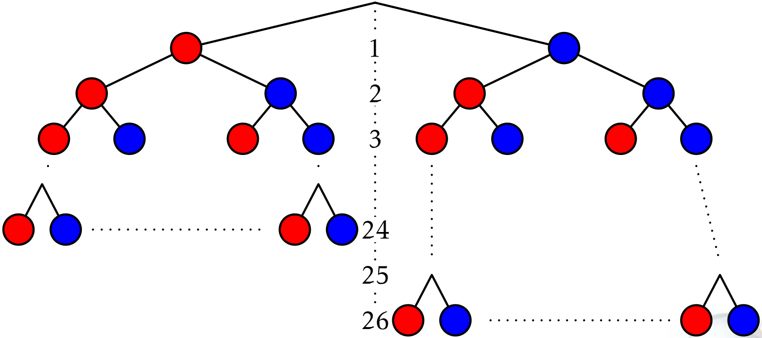
- On peut utiliser moins de jetons pour certaines images, mais il en faut donc plus pour d'autres.



- C'est le pire des cas, ou le cas moyen, qu'il faut étudier.
- On se rapproche des algorithmes de compression.

Code "déséquilibré"

- On peut utiliser moins de jetons pour certaines images, mais il en faut donc plus pour d'autres.



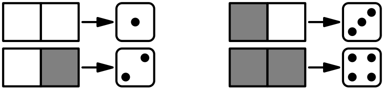
- C'est le pire des cas, ou le cas moyen, qu'il faut étudier.
- On se rapproche des algorithmes de compression.
- En général, on préfère des codes équilibrés avec une taille constante.

Avec des dés

- Il est possible de transmettre une image avec 25 dés.
- Comment faire mieux ?

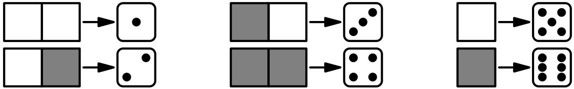
Avec des dés

- Il est possible de transmettre une image avec 25 dés.
- Comment faire mieux ?
- Coder 2 pixels avec un dé :



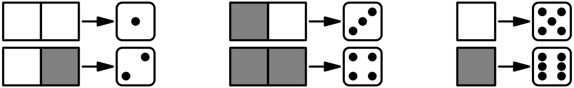
Avec des dés

- Il est possible de transmettre une image avec 25 dés.
- Comment faire mieux ?
- Coder 2 pixels avec un dé :



Avec des dés

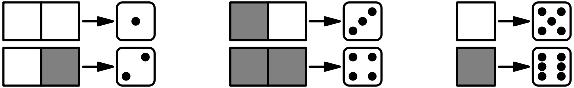
- Il est possible de transmettre une image avec 25 dés.
- Comment faire mieux ?
- Coder 2 pixels avec un dé :



- Il faut 13 dés.

Avec des dés

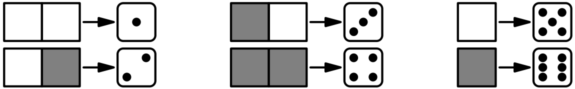
- Il est possible de transmettre une image avec 25 dés.
- Comment faire mieux ?
- Coder 2 pixels avec un dé :



- Il faut 13 dés.
- Nombre minimal de dés :

Avec des dés

- Il est possible de transmettre une image avec 25 dés.
- Comment faire mieux ?
- Coder 2 pixels avec un dé :

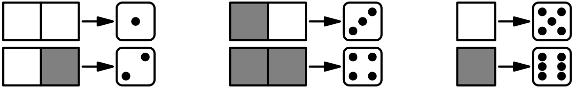


- Il faut 13 dés.
- Nombre minimal de dés :
  - On cherche le plus petit $n \in \mathbb{N}$ tel que :

$$6^n > 2^{25}$$

Avec des dés

- Il est possible de transmettre une image avec 25 dés.
- Comment faire mieux ?
- Coder 2 pixels avec un dé :



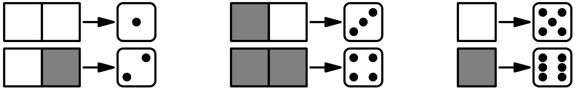
- Il faut 13 dés.
- Nombre minimal de dés :
  - On cherche le plus petit $n \in \mathbb{N}$ tel que :

$$6^n > 2^{25}$$

  - La réponse est $6^{10} = 60\,466\,176$.

Avec des dés

- Il est possible de transmettre une image avec 25 dés.
- Comment faire mieux ?
- Coder 2 pixels avec un dé :



- Il faut 13 dés.
- Nombre minimal de dés :
  - On cherche le plus petit $n \in \mathbb{N}$ tel que :

$$6^n > 2^{25}$$

  - La réponse est $6^{10} = 60\,466\,176$.
- Comment faire avec 10 dés ?

Compter en base 2, 10 et 6

En binaire	En décimal	En base 6
	0	
	1	
	2	
	3	
	4	
	5	
	6	
	7	
	8	
	9	

Compter en base 2, 10 et 6

En binaire	En décimal	En base 6
	0	
	1	
	2	
	3	
	4	
	5	
	6	
	7	
	8	
	9	
	10	

Compter en base 2, 10 et 6

En binaire	En décimal	En base 6
	0	
	1	
	2	
	3	
	4	
	5	
	6	
	7	
	8	
	9	
	10	
	11	

Compter en base 2, 10 et 6

En binaire	En décimal	En base 6
	0	
	1	
	2	
	3	
	4	
	5	
	6	
	7	
	8	
	9	
	10	
	11	
	⋮	

Compter en base 2, 10 et 6

En binaire	En décimal	En base 6
0	0	
1	1	
	2	
	3	
	4	
	5	
	6	
	7	
	8	
	9	
	10	
	11	
	⋮	

Compter en base 2, 10 et 6

En binaire	En décimal	En base 6
0	0	
1	1	
10	2	
	3	
	4	
	5	
	6	
	7	
	8	
	9	
	10	
	11	
	⋮	

Compter en base 2, 10 et 6

En binaire	En décimal	En base 6
0	0	
1	1	
10	2	
11	3	
	4	
	5	
	6	
	7	
	8	
	9	
	10	
	11	
	⋮	

Compter en base 2, 10 et 6

En binaire	En décimal	En base 6
0	0	
1	1	
10	2	
11	3	
100	4	
	5	
	6	
	7	
	8	
	9	
	10	
	11	
	⋮	

Compter en base 2, 10 et 6

En binaire	En décimal	En base 6
0	0	
1	1	
10	2	
11	3	
100	4	
101	5	
110	6	
111	7	
1000	8	
1001	9	
1010	10	
1011	11	
⋮	⋮	

Compter en base 2, 10 et 6

En binaire	En décimal	En base 6
0	0	0
1	1	1
10	2	2
11	3	3
100	4	4
101	5	5
110	6	
111	7	
1000	8	
1001	9	
1010	10	
1011	11	
⋮	⋮	

Compter en base 2, 10 et 6

En binaire	En décimal	En base 6
0	0	0
1	1	1
10	2	2
11	3	3
100	4	4
101	5	5
110	6	10
111	7	
1000	8	
1001	9	
1010	10	
1011	11	
⋮	⋮	

Compter en base 2, 10 et 6

En binaire	En décimal	En base 6
0	0	0
1	1	1
10	2	2
11	3	3
100	4	4
101	5	5
110	6	10
111	7	11
1000	8	
1001	9	
1010	10	
1011	11	
⋮	⋮	

Compter en base 2, 10 et 6

En binaire	En décimal	En base 6
0	0	0
1	1	1
10	2	2
11	3	3
100	4	4
101	5	5
110	6	10
111	7	11
1000	8	12
1001	9	13
1010	10	14
1011	11	15
⋮	⋮	⋮

Compter en base 2, 10 et 6

En binaire	En décimal	En base 6
0	0	0
1	1	1
10	2	2
11	3	3
100	4	4
101	5	5
110	6	10
111	7	11
1000	8	12
1001	9	13
1010	10	14
1011	11	15
⋮	⋮	⋮
11111		

Compter en base 2, 10 et 6

En binaire	En décimal	En base 6
0	0	0
1	1	1
10	2	2
11	3	3
100	4	4
101	5	5
110	6	10
111	7	11
1000	8	12
1001	9	13
1010	10	14
1011	11	15
⋮	⋮	⋮
11111	31	

Compter en base 2, 10 et 6

En binaire	En décimal	En base 6
0	0	0
1	1	1
10	2	2
11	3	3
100	4	4
101	5	5
110	6	10
111	7	11
1000	8	12
1001	9	13
1010	10	14
1011	11	15
⋮	⋮	⋮
11111	31	51

Compter en base 2, 10 et 6

En binaire	En décimal	En base 6
00000	00	00
00001	01	01
00010	02	02
00011	03	03
00100	04	04
00101	05	05
00110	06	10
00111	07	11
01000	08	12
01001	09	13
01010	10	14
01011	11	15
⋮	⋮	⋮
11111	31	51

Du binaire à la base 6

- En décimal :

$$243 = 2\times 10^2 + 4\times 10^1 + 3\times 10^0$$

Du binaire à la base 6

- En décimal :

$$243 = 2\times 10^2 + 4\times 10^1 + 3\times 10^0$$

- En binaire, c'est pareil :

$$10011_2 = 1\times 2^4 + 0\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0 = 16 + 2 + 1 = 19$$

Du binaire à la base 6

- En décimal :

$$243 = 2\times10^2 + 4\times10^1 + 3\times10^0$$

- En binaire, c'est pareil :

$$10011_2 = 1\times2^4 + 0\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 = 16 + 2 + 1 = 19$$

- Pour convertir en base 6, on cherche a et $b \in \{0,1,2,3,4,5\}$ tels que :

$$19 = a\times6^1 + b\times6^0 = 6a + b$$

Du binaire à la base 6

- En décimal :

$$243 = 2\times10^2 + 4\times10^1 + 3\times10^0$$

- En binaire, c'est pareil :

$$10011_2 = 1\times2^4 + 0\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 = 16 + 2 + 1 = 19$$

- Pour convertir en base 6, on cherche a et $b \in \{0,1,2,3,4,5\}$ tels que :

$$19 = a\times6^1 + b\times6^0 = 6a + b$$

- a est le dividende et b le reste de la division euclidienne de 19 par 6.

Du binaire à la base 6

- En décimal :

$$243 = 2\times10^2 + 4\times10^1 + 3\times10^0$$

- En binaire, c'est pareil :

$$10011_2 = 1\times2^4 + 0\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 = 16 + 2 + 1 = 19$$

- Pour convertir en base 6, on cherche a et $b \in \{0,1,2,3,4,5\}$ tels que :

$$19 = a\times6^1 + b\times6^0 = 6a + b$$

- a est le dividende et b le reste de la division euclidienne de 19 par 6.
- Donc $a = 3$ et $b = 1$.

Du binaire à la base 6

- En décimal :

$$243 = 2\times10^2 + 4\times10^1 + 3\times10^0$$

- En binaire, c'est pareil :

$$10011_2 = 1\times2^4 + 0\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 = 16 + 2 + 1 = 19$$

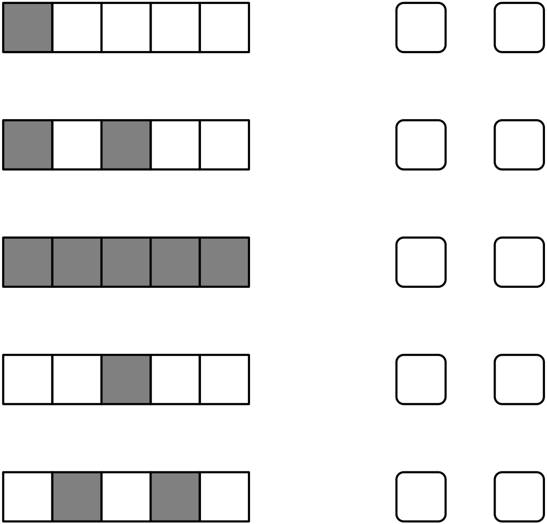
- Pour convertir en base 6, on cherche a et $b \in \{0,1,2,3,4,5\}$ tels que :

$$19 = a\times6^1 + b\times6^0 = 6a + b$$

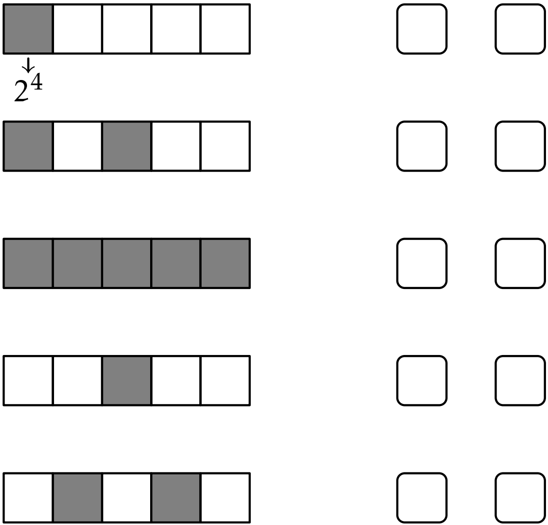
- a est le dividende et b le reste de la division euclidienne de 19 par 6.
- Donc $a = 3$ et $b = 1$.

binaire		décimal		base 6
10011	=	19	=	31

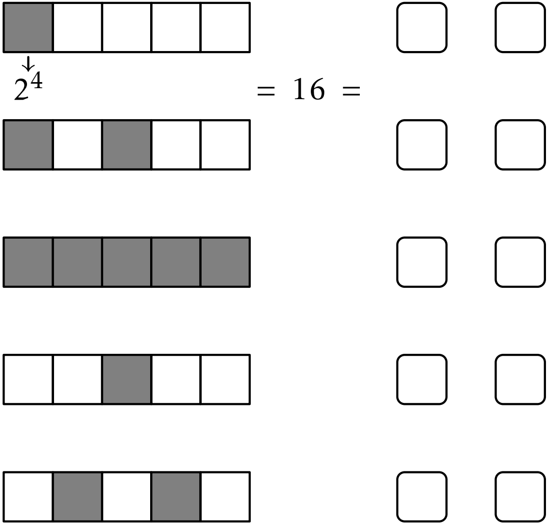
Avec 10 dés



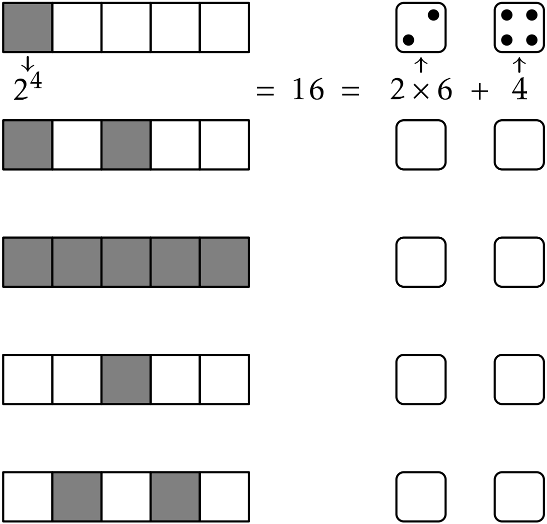
Avec 10 dés



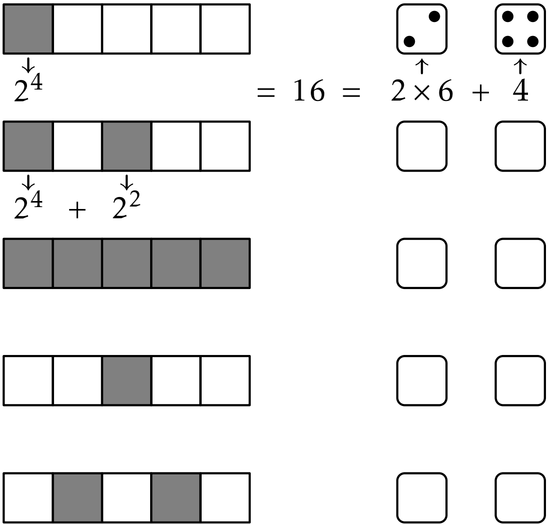
Avec 10 dés



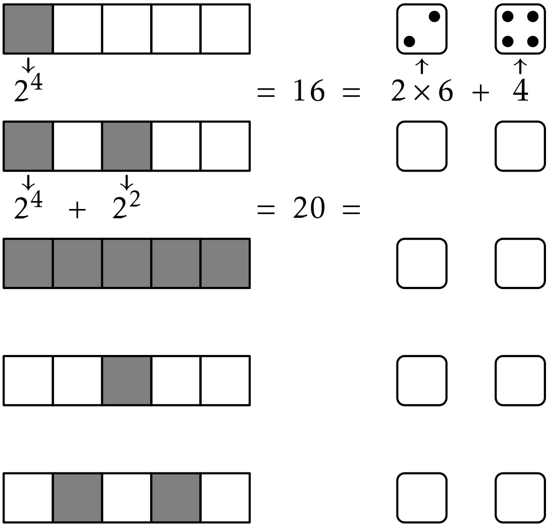
Avec 10 dés



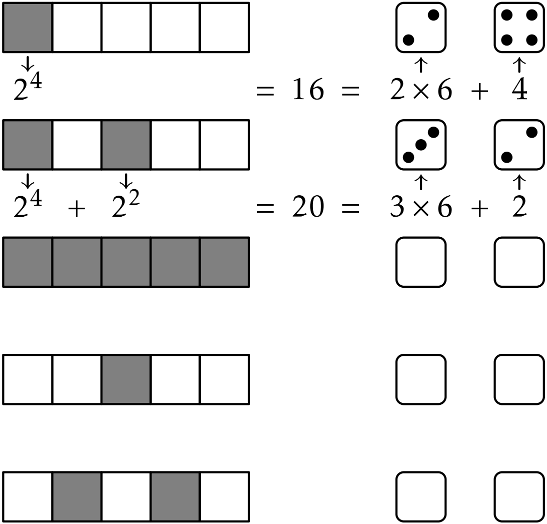
Avec 10 dés



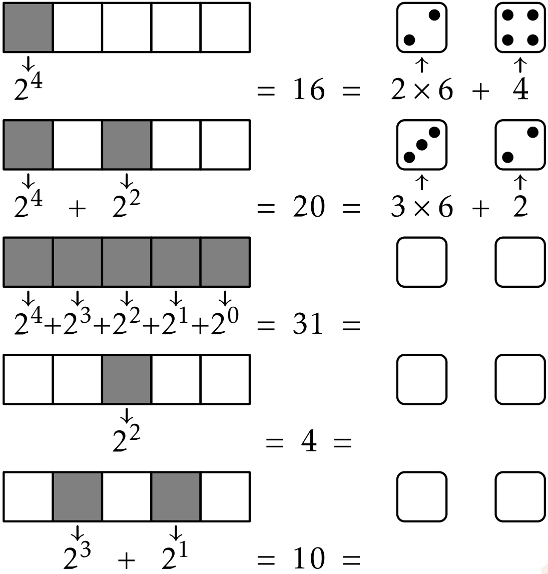
Avec 10 dés



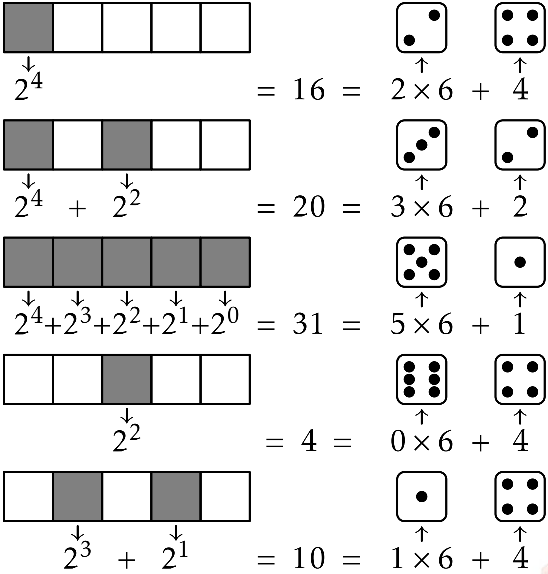
Avec 10 dés



Avec 10 dés



Avec 10 dés



$$2^4 = 16 = 2 \times 6 + 4$$

$$2^4 + 2^2 = 20 = 3 \times 6 + 2$$

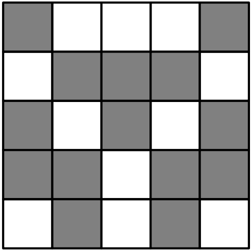
$$2^4+2^3+2^2+2^1+2^0 = 31 = 5 \times 6 + 1$$

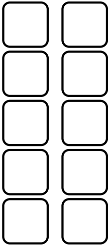
$$2^2 = 4 = 0 \times 6 + 4$$

$$2^3 + 2^1 = 10 = 1 \times 6 + 4$$

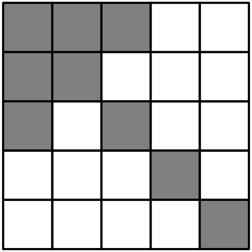
À vous de jouer

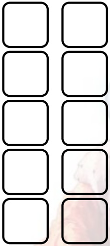
16 8 4 2 1



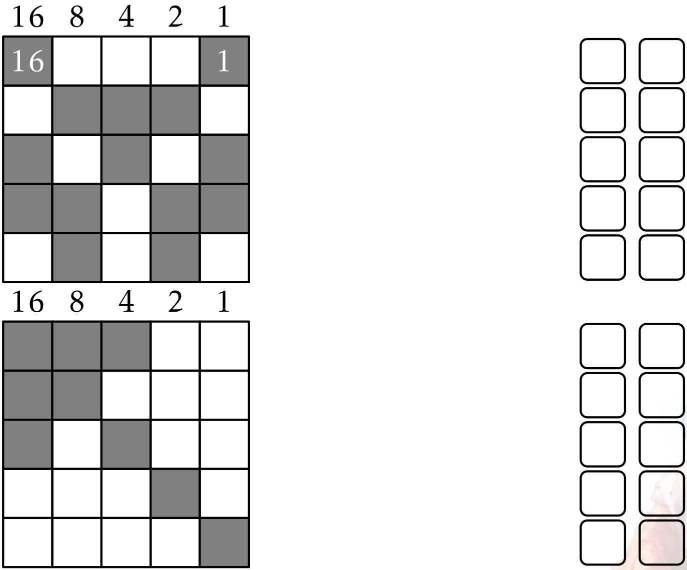


16 8 4 2 1

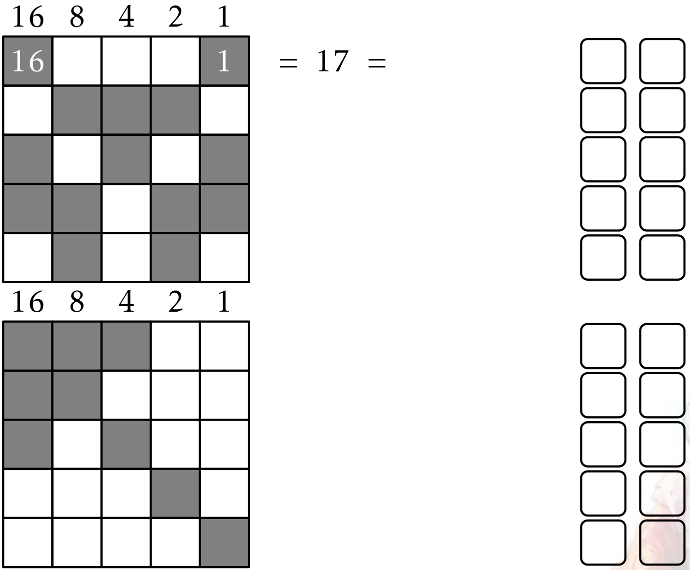




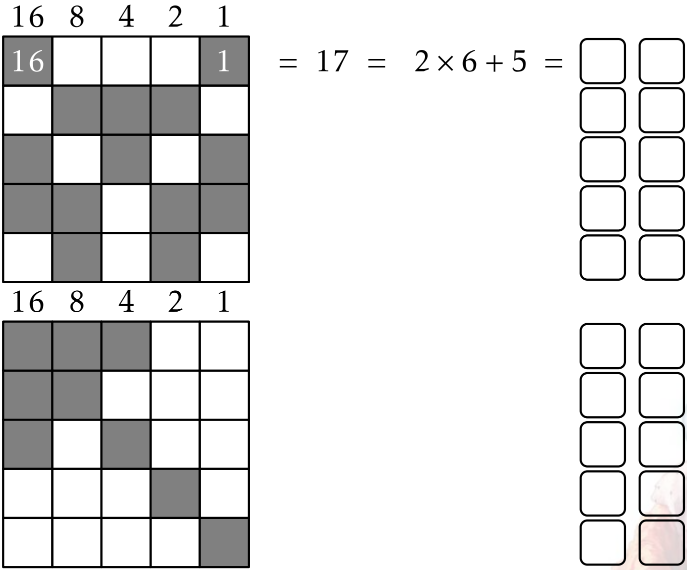
À vous de jouer



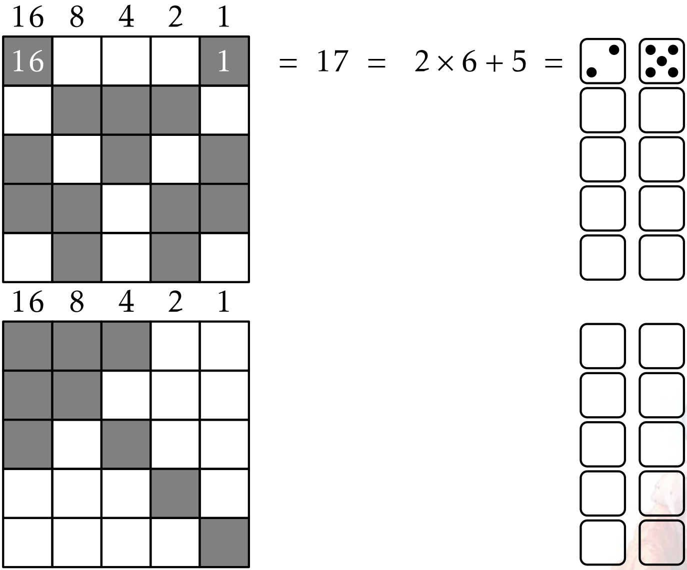
À vous de jouer



À vous de jouer



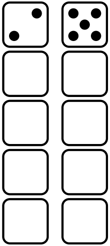
À vous de jouer



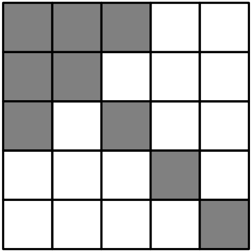
À vous de jouer

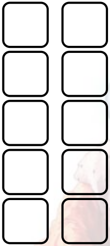
16	8	4	2	1
16				1
	8	4	2	
16		4		1
16	8		2	1
	8		2	

$= 17 = 2 \times 6 + 5 =$



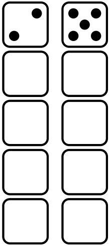
16 8 4 2 1



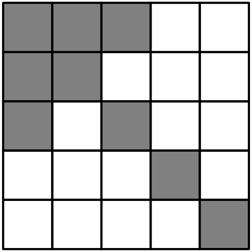


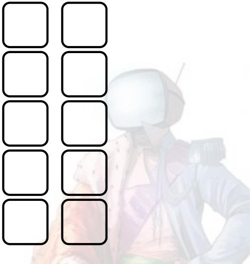
À vous de jouer

16	8	4	2	1		
16				1	$= 17 =$	$2 \times 6 + 5 =$
	8	4	2		$= 14 =$	
16		4		1	$= 21 =$	
16	8		2	1	$= 27 =$	
	8		2		$= 10 =$	

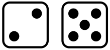


16 8 4 2 1

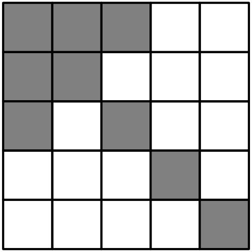




À vous de jouer

16	8	4	2	1		
16				1	$= 17 = 2 \times 6 + 5 =$	
	8	4	2		$= 14 = 2 \times 6 + 2 =$	
16		4		1	$= 21 = 3 \times 6 + 3 =$	
16	8		2	1	$= 27 = 4 \times 6 + 3 =$	
	8		2		$= 10 = 1 \times 6 + 4 =$	

16 8 4 2 1



À vous de jouer

16	8	4	2	1
16				1
	8	4	2	
16		4		1
16	8		2	1
	8		2	

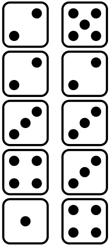
$= 17 = 2 \times 6 + 5 =$

$= 14 = 2 \times 6 + 2 =$

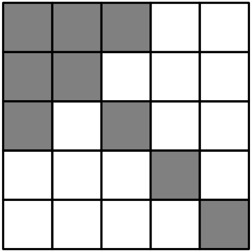
$= 21 = 3 \times 6 + 3 =$

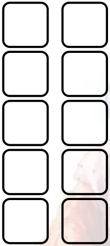
$= 27 = 4 \times 6 + 3 =$

$= 10 = 1 \times 6 + 4 =$



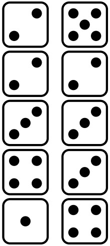
16 8 4 2 1



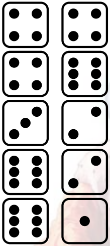


À vous de jouer

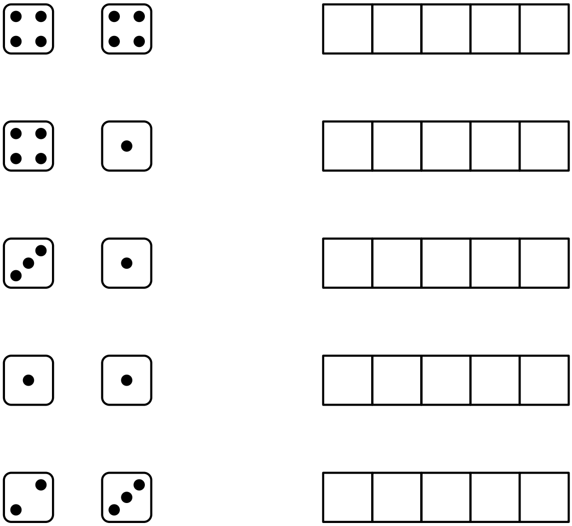
16	8	4	2	1				
16				1	$= 17$	$= 2\times 6+5$	=
	8	4	2		$= 14$	$= 2\times 6+2$	=
16		4		1	$= 21$	$= 3\times 6+3$	=
16	8		2	1	$= 27$	$= 4\times 6+3$	=
	8		2		$= 10$	$= 1\times 6+4$	=



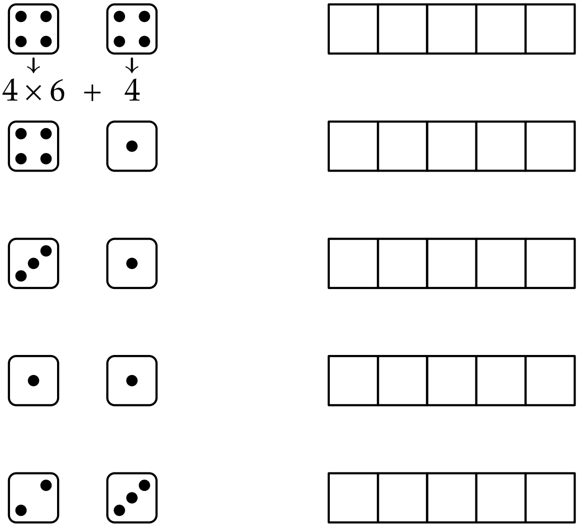
16	8	4	2	1				
16	8	4			$= 28$	$= 4\times 6+4$	=
16	8				$= 24$	$= 4\times 6+0$	=
16		4			$= 20$	$= 3\times 6+2$	=
			2		$= 2$	$= 0\times 6+2$	=
				1	$= 1$	$= 0\times 6+1$	=



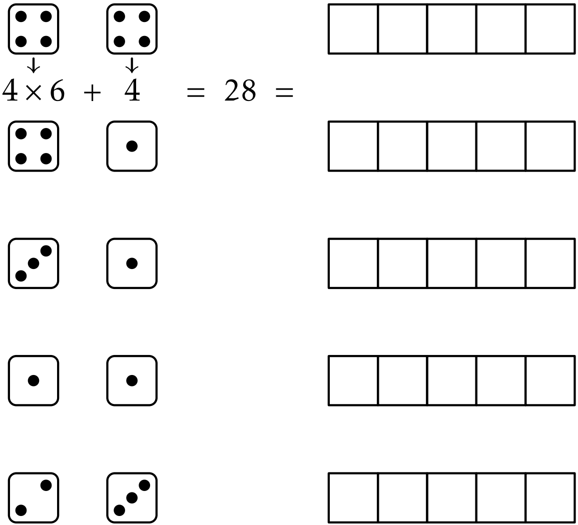
Pour décoder



Pour décoder

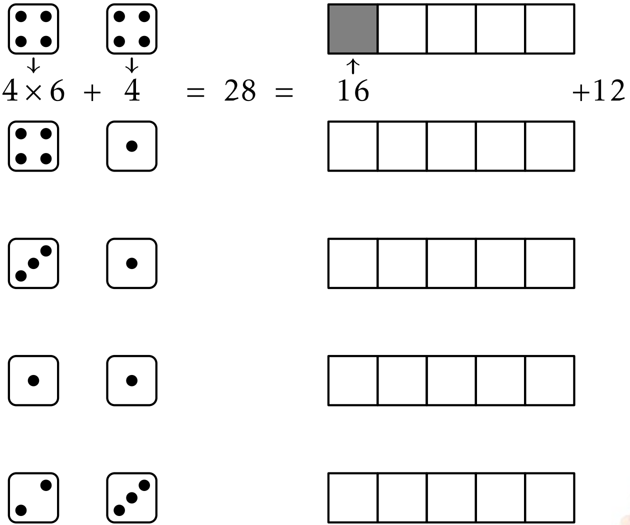


Pour décoder

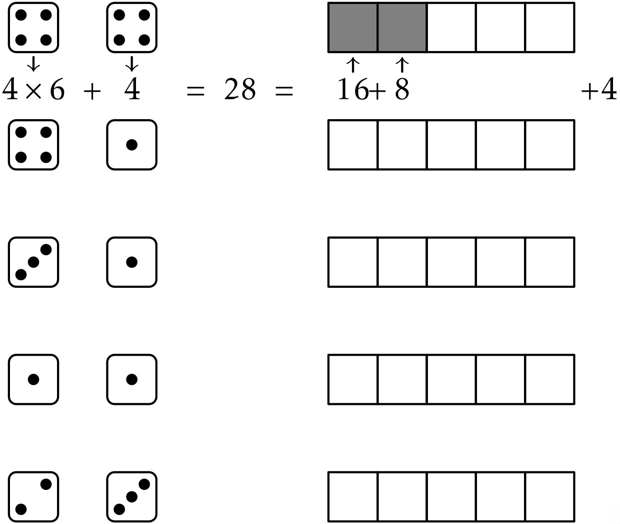


$$4 \times 6 + 4 = 28 =$$

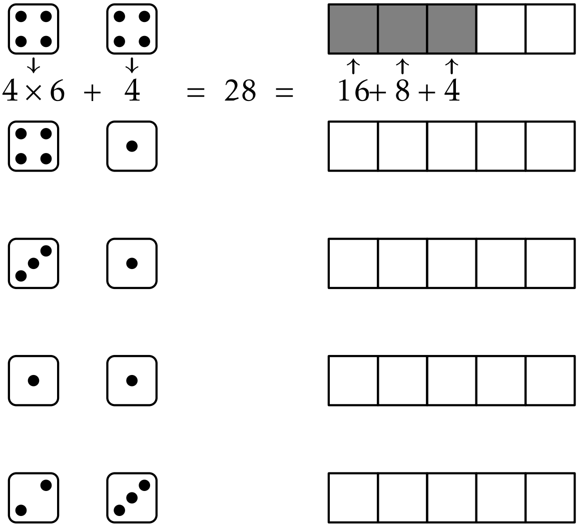
Pour décoder



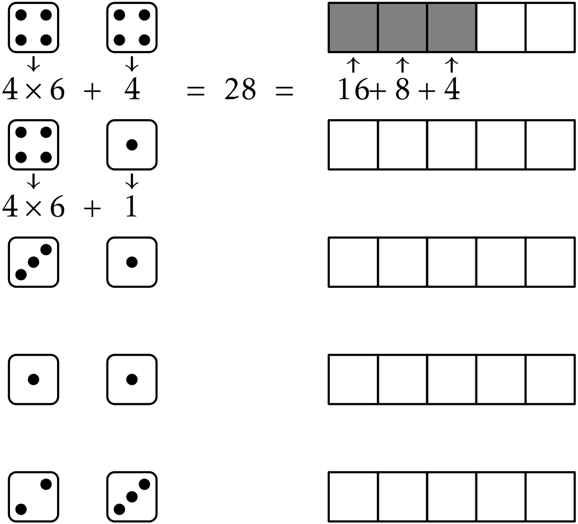
Pour décoder



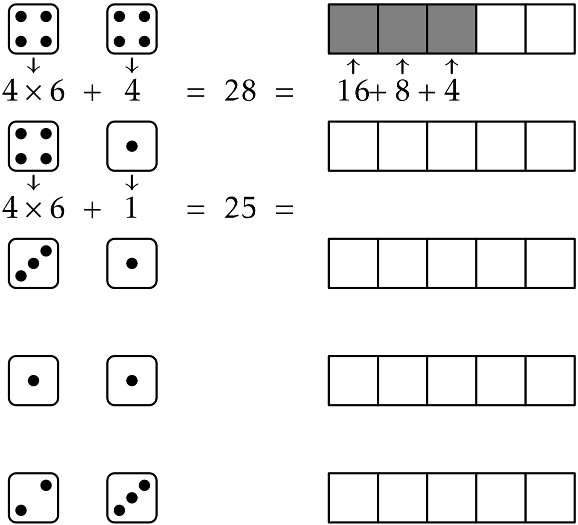
Pour décoder



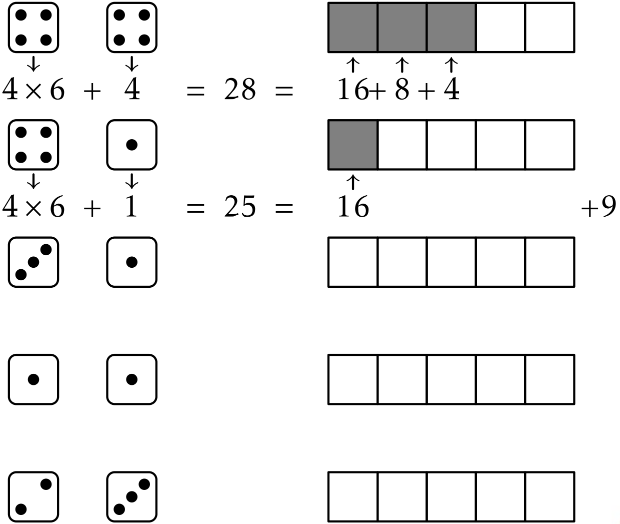
Pour décoder



Pour décoder



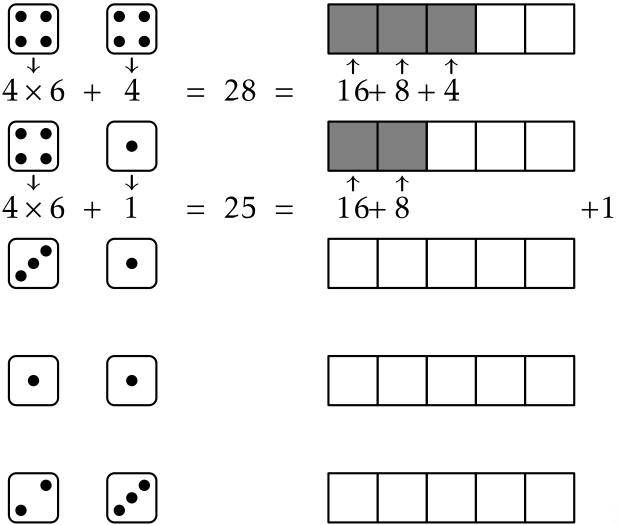
Pour décoder



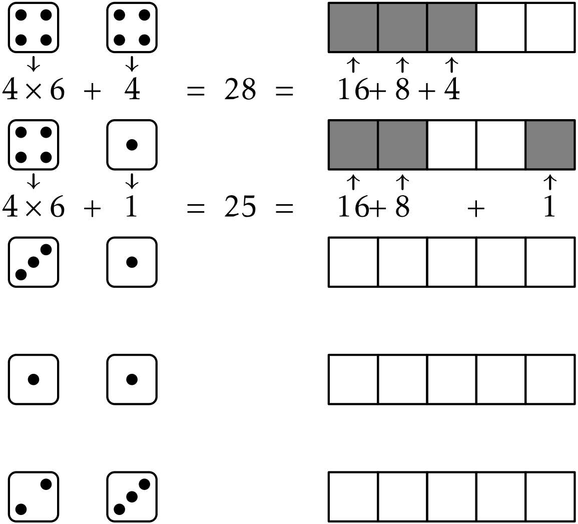
$$4 \times 6 + 4 = 28 = 16 + 8 + 4$$

$$4 \times 6 + 1 = 25 = 16 \quad +9$$

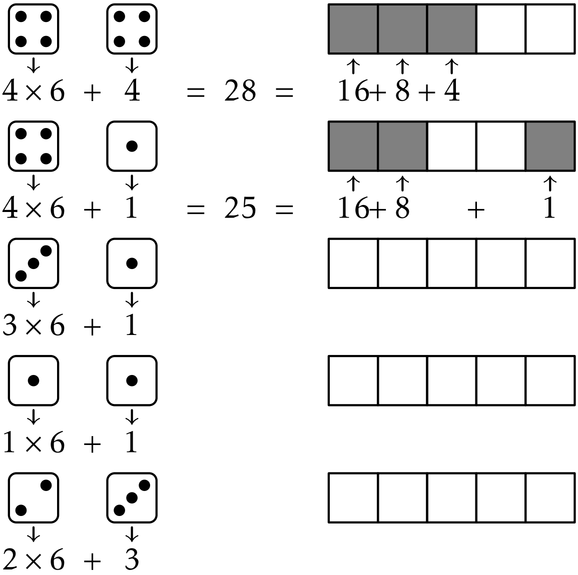
Pour décoder



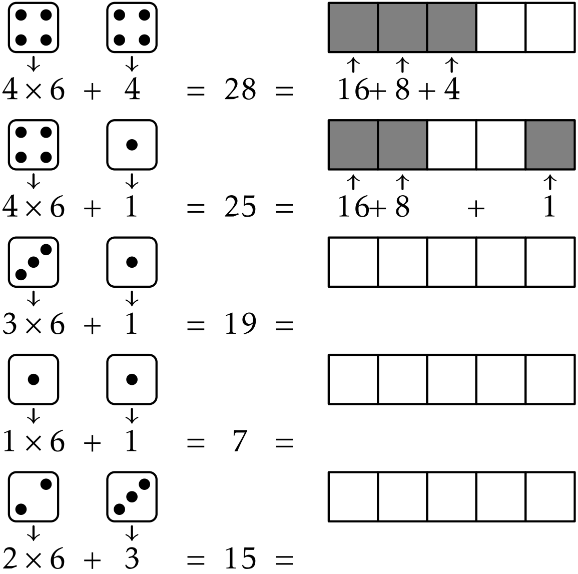
Pour décoder



Pour décoder



Pour décoder



$$4 \times 6 + 4 = 28 = 16 + 8 + 4$$

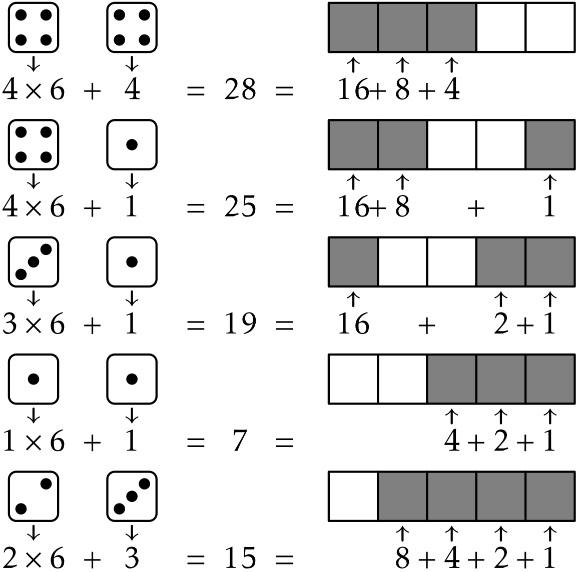
$$4 \times 6 + 1 = 25 = 16 + 8 + 1$$

$$3 \times 6 + 1 = 19 =$$

$$1 \times 6 + 1 = 7 =$$

$$2 \times 6 + 3 = 15 =$$

Pour décoder



$$4 \times 6 + 4 = 28 = 16 + 8 + 4$$

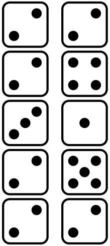
$$4 \times 6 + 1 = 25 = 16 + 8 + 1$$

$$3 \times 6 + 1 = 19 = 16 + 2 + 1$$

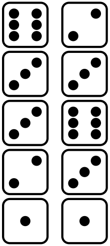
$$1 \times 6 + 1 = 7 = 4 + 2 + 1$$

$$2 \times 6 + 3 = 15 = 8 + 4 + 2 + 1$$

À vous de jouer

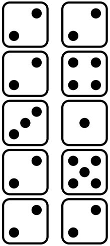


16	8	4	2	1



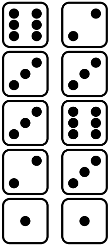
16	8	4	2	1

À vous de jouer



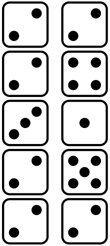
$= 2 \times 6 + 2$

16	8	4	2	1



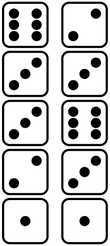
16	8	4	2	1

À vous de jouer



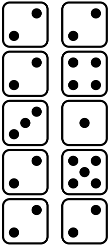
$= 2 \times 6 + 2 \;=\; 14 \;=$

16	8	4	2	1



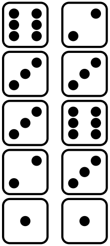
16	8	4	2	1

À vous de jouer



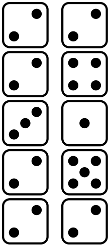
$$= 2 \times 6 + 2 = 14 =$$

16	8	4	2	1	
	8				+6



16	8	4	2	1

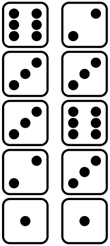
À vous de jouer



$= 2 \times 6 + 2 \;=\; 14 \;=$

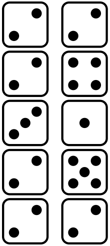
16	8	4	2	1
	8	4		

$+2$



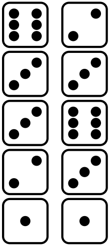
16	8	4	2	1

À vous de jouer



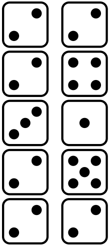
$= 2 \times 6 + 2 \;=\; 14 \;=$

16	8	4	2	1
	8	4	2	



16	8	4	2	1

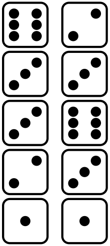
À vous de jouer



$$= 2 \times 6 + 2 \;=\; 14 \;=$$

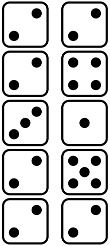
$$= 2 \times 6 + 4$$

16	8	4	2	1
	8	4	2	



16	8	4	2	1

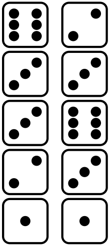
À vous de jouer



$= 2 \times 6 + 2 = 14 =$

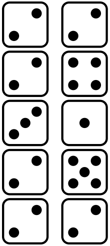
$= 2 \times 6 + 4 = 16 =$

16	8	4	2	1
	8	4	2	



16	8	4	2	1

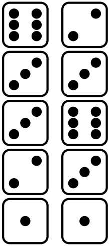
À vous de jouer



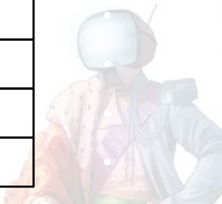
$$= 2 \times 6 + 2 \;=\; 14 \;=$$

$$= 2 \times 6 + 4 \;=\; 16 \;=$$

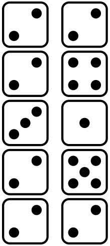
16	8	4	2	1
	8	4	2	
16				



16	8	4	2	1



À vous de jouer



$= 2 \times 6 + 2 \; = \; 14 \; =$

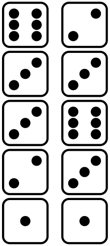
$= 2 \times 6 + 4 \; = \; 16 \; =$

$= 3 \times 6 + 1$

$= 2 \times 6 + 5$

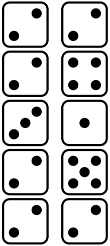
$= 2 \times 6 + 2$

16	8	4	2	1
	8	4	2	
16				



16	8	4	2	1

À vous de jouer



$= 2 \times 6 + 2 = 14 =$

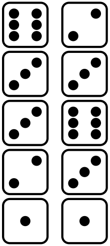
$= 2 \times 6 + 4 = 16 =$

$= 3 \times 6 + 1 = 19 =$

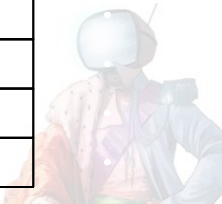
$= 2 \times 6 + 5 = 17 =$

$= 2 \times 6 + 2 = 14 =$

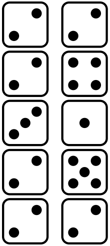
16	8	4	2	1
	8	4	2	
16				



16	8	4	2	1



À vous de jouer



$= 2 \times 6 + 2 = 14 =$

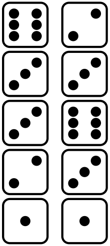
$= 2 \times 6 + 4 = 16 =$

$= 3 \times 6 + 1 = 19 =$

$= 2 \times 6 + 5 = 17 =$

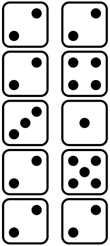
$= 2 \times 6 + 2 = 14 =$

16	8	4	2	1	
	8	4	2		
16					
16					+3
16					+1
	8				+6



16	8	4	2	1

À vous de jouer



$= 2 \times 6 + 2 = 14 =$

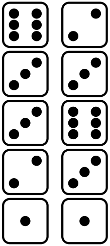
$= 2 \times 6 + 4 = 16 =$

$= 3 \times 6 + 1 = 19 =$

$= 2 \times 6 + 5 = 17 =$

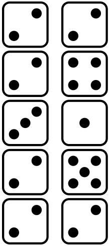
$= 2 \times 6 + 2 = 14 =$

16	8	4	2	1
	8	4	2	
16				
16			2	1
16				1
	8	4	2	

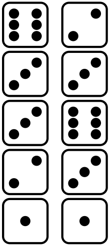


16	8	4	2	1

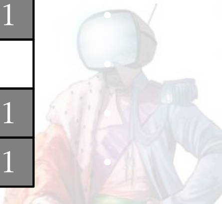
À vous de jouer



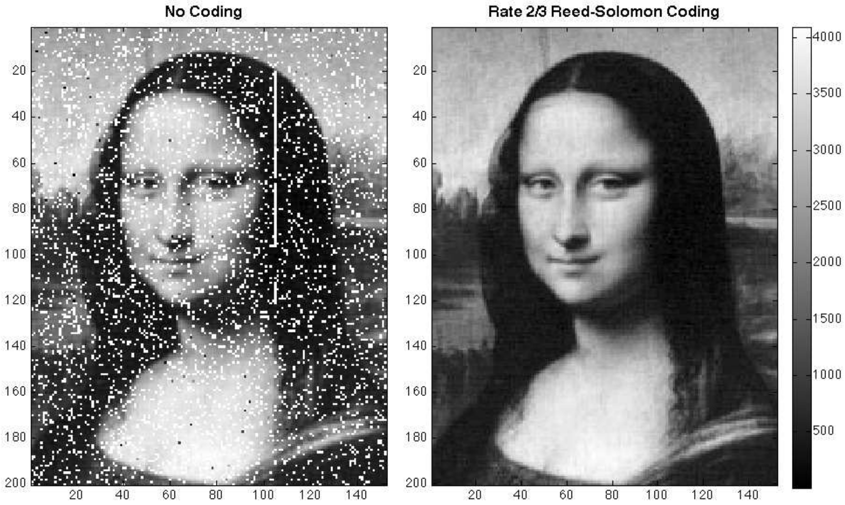
	16	8	4	2	1
$= 2 \times 6 + 2 = 14 =$		8	4	2	
$= 2 \times 6 + 4 = 16 =$	16				
$= 3 \times 6 + 1 = 19 =$	16			2	1
$= 2 \times 6 + 5 = 17 =$	16				1
$= 2 \times 6 + 2 = 14 =$		8	4	2	



	16	8	4	2	1
$= 0 \times 6 + 2 = 2 =$				2	
$= 3 \times 6 + 3 = 21 =$	16		4		1
$= 3 \times 6 + 0 = 18 =$	16			2	
$= 2 \times 6 + 3 = 15 =$		8	4	2	1
$= 1 \times 6 + 1 = 7 =$			4	2	1



Comment corriger les erreurs ?



Détecter l'erreur

- Idée naïve : envoyer 2 fois le message.

Détecter l'erreur

- Idée naïve : envoyer 2 fois le message.
- Il suffit de rajouter un **bit de parité**.
- Dans notre cas : rajouter un jeton de la couleur ayant un nombre impair de pixels.

Détecter l'erreur

- Idée naïve : envoyer 2 fois le message.
- Il suffit de rajouter un **bit de parité**.
- Dans notre cas : rajouter un jeton de la couleur ayant un nombre impair de pixels.
- En pratique : somme des bits du message, modulo 2.
- On ajoute le bit obtenu à la fin du message.
- Si on change un bit du message, on change la parité du total.

Détecter l'erreur

- Idée naïve : envoyer 2 fois le message.
- Il suffit de rajouter un **bit de parité**.
- Dans notre cas : rajouter un jeton de la couleur ayant un nombre impair de pixels.
- En pratique : somme des bits du message, modulo 2.
- On ajoute le bit obtenu à la fin du message.
- Si on change un bit du message, on change la parité du total.
- Détecte un nombre impair d'erreurs, mais pas un nombre pair.

Détecter l'erreur

- Idée naïve : envoyer 2 fois le message.
- Il suffit de rajouter un **bit de parité**.
- Dans notre cas : rajouter un jeton de la couleur ayant un nombre impair de pixels.
- En pratique : somme des bits du message, modulo 2.
- On ajoute le bit obtenu à la fin du message.
- Si on change un bit du message, on change la parité du total.
- Détecte un nombre impair d'erreurs, mais pas un nombre pair.
- Ne permet pas de savoir où est l'erreur.

Détecter l'erreur

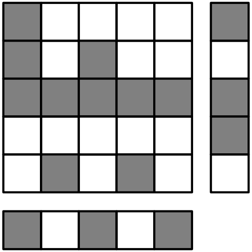
- Idée naïve : envoyer 2 fois le message.
- Il suffit de rajouter un **bit de parité**.
- Dans notre cas : rajouter un jeton de la couleur ayant un nombre impair de pixels.
- En pratique : somme des bits du message, modulo 2.
- On ajoute le bit obtenu à la fin du message.
- Si on change un bit du message, on change la parité du total.
- Détecte un nombre impair d'erreurs, mais pas un nombre pair.
- Ne permet pas de savoir où est l'erreur.
- Chiffres de contrôle : code barre, carte bleue, numéro INSEE…

Corriger l'erreur

- Idée naïve : envoyer 3 fois le message.

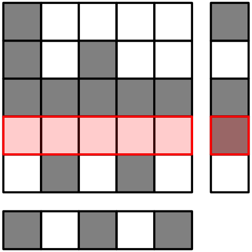
Corriger l'erreur

- Idée naïve : envoyer 3 fois le message.
- Rajouter un bit de parité par ligne et par colonne :



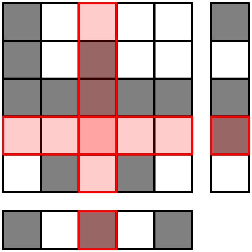
Corriger l'erreur

- Idée naïve : envoyer 3 fois le message.
- Rajouter un bit de parité par ligne et par colonne :



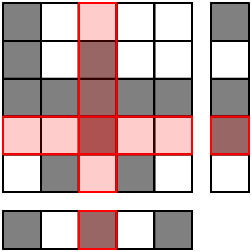
Corriger l'erreur

- Idée naïve : envoyer 3 fois le message.
- Rajouter un bit de parité par ligne et par colonne :



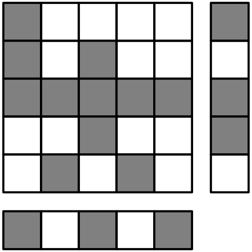
Corriger l'erreur

- Idée naïve : envoyer 3 fois le message.
- Rajouter un bit de parité par ligne et par colonne :



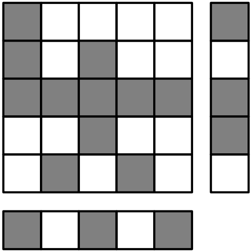
Corriger l'erreur

- Idée naïve : envoyer 3 fois le message.
- Rajouter un bit de parité par ligne et par colonne :



Corriger l'erreur

- Idée naïve : envoyer 3 fois le message.
- Rajouter un bit de parité par ligne et par colonne :



- Peut-on faire mieux ?

Distance de Hamming

- Nombre de symboles différents entre deux “mots” de même longueurs.

Distance de Hamming

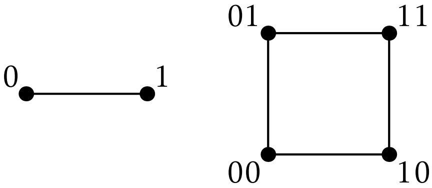
- Nombre de symboles différents entre deux “mots” de même longueurs.
- La distance entre 11011 et 11000 est de 2.

Distance de Hamming

- Nombre de symboles différents entre deux “mots” de même longueurs.
- La distance entre 11011 et 11000 est de 2.
- On représente souvent les mots possibles à l'aide d'un graphe :

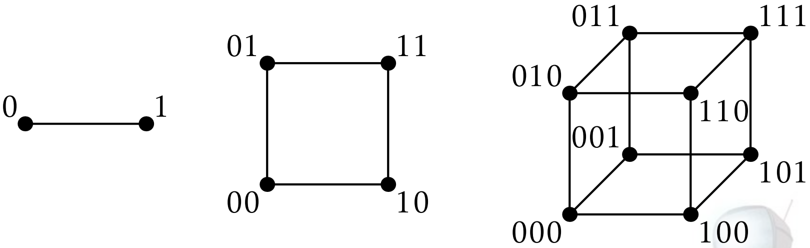
Distance de Hamming

- Nombre de symboles différents entre deux “mots” de même longueurs.
- La distance entre 11011 et 11000 est de 2.
- On représente souvent les mots possibles à l'aide d'un graphe :



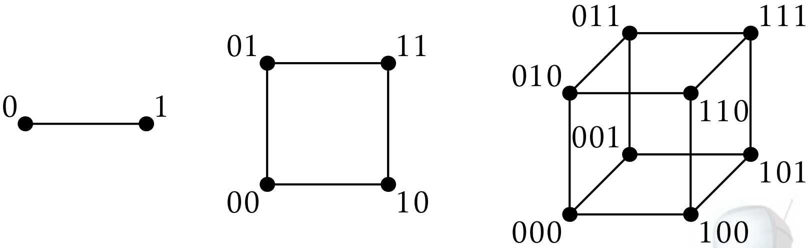
Distance de Hamming

- Nombre de symboles différents entre deux “mots” de même longueurs.
- La distance entre 11011 et 11000 est de 2.
- On représente souvent les mots possibles à l'aide d'un graphe :



Distance de Hamming

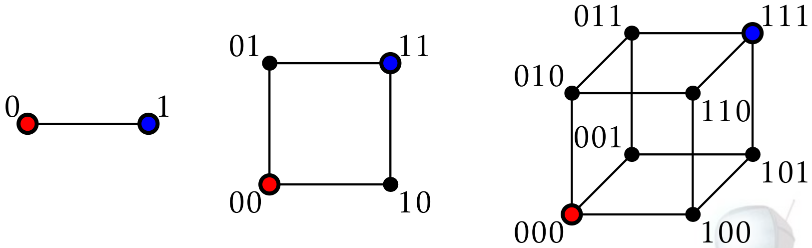
- Nombre de symboles différents entre deux “mots” de même longueurs.
- La distance entre 11011 et 11000 est de 2.
- On représente souvent les mots possibles à l'aide d'un graphe :



- Pour corriger une erreur, il faut que :

Distance de Hamming

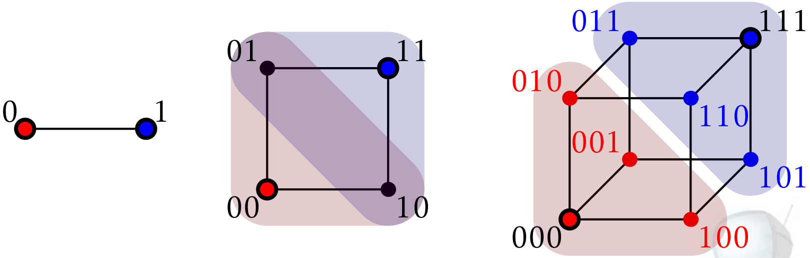
- Nombre de symboles différents entre deux “mots” de même longueurs.
- La distance entre 11011 et 11000 est de 2.
- On représente souvent les mots possibles à l'aide d'un graphe :



- Pour corriger une erreur, il faut que :
  - le mot reçu ne soit pas un des mots du code

Distance de Hamming

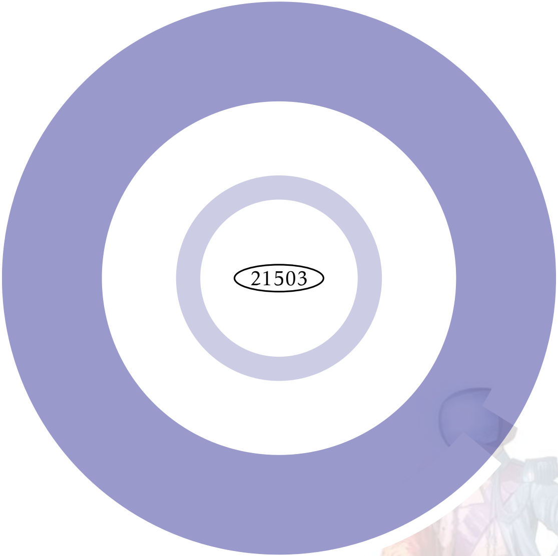
- Nombre de symboles différents entre deux “mots” de même longueurs.
- La distance entre 11011 et 11000 est de 2.
- On représente souvent les mots possibles à l'aide d'un graphe :



- Pour corriger une erreur, il faut que :
  - le mot reçu ne soit pas un des mots du code
  - il soit plus près d'un des mots du code que des autres

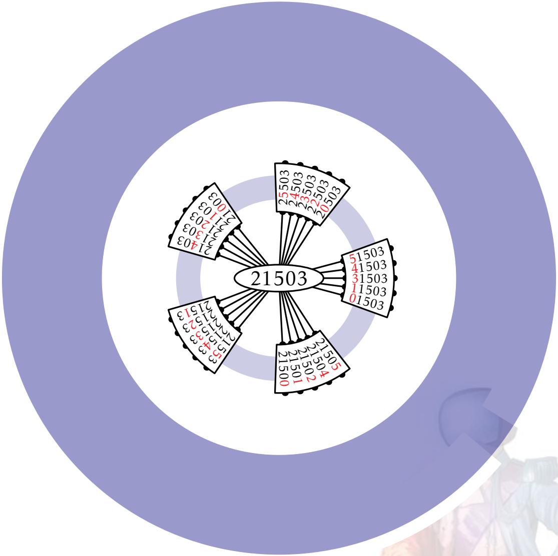
Boule de Hamming

- Alphabet de taille 6.
- Mots de longueur 5.
- À distance 1 :



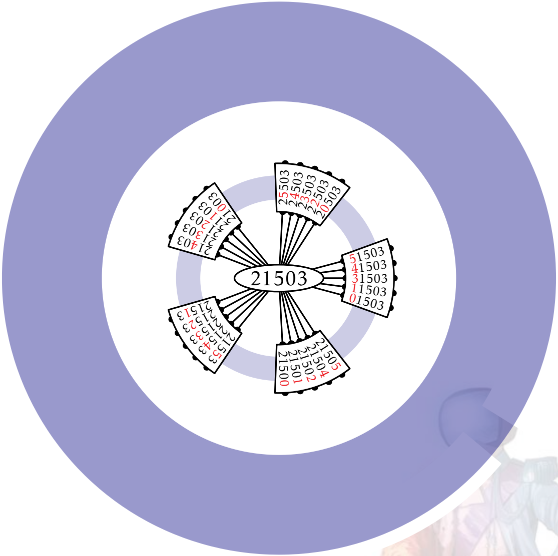
Boule de Hamming

- Alphabet de taille 6.
- Mots de longueur 5.
- À distance 1 : $\binom{5}{1} \times 5^1 = 25$



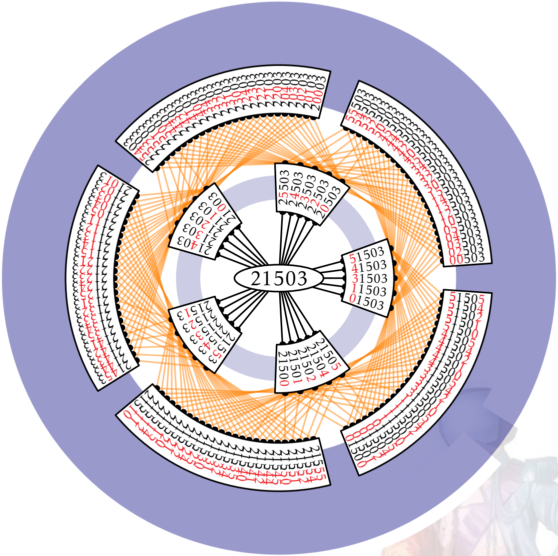
Boule de Hamming

- Alphabet de taille 6.
- Mots de longueur 5.
- À distance 1 : $\binom{5}{1} \times 5^1 = 25$
- À distance 2 :



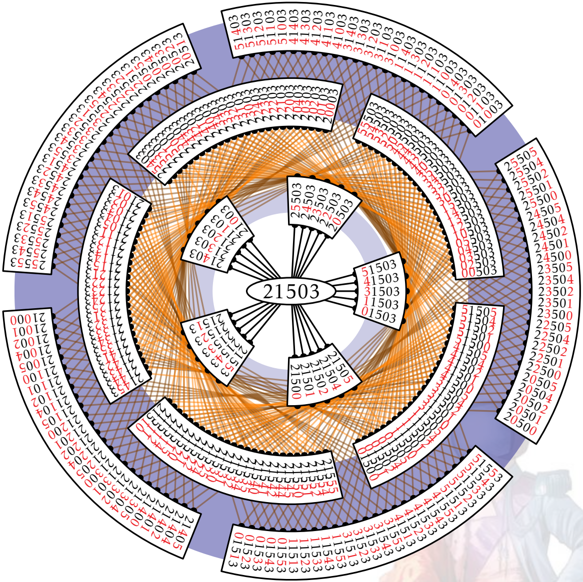
Boule de Hamming

- Alphabet de taille 6.
- Mots de longueur 5.
- À distance 1 : $\binom{5}{1} \times 5^1 = 25$
- À distance 2 :



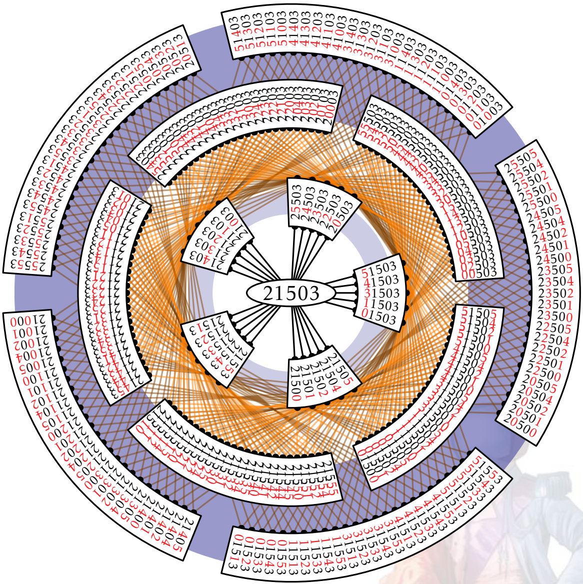
Boule de Hamming

- Alphabet de taille 6.
- Mots de longueur 5.
- À distance 1 : $\binom{5}{1} \times 5^1 = 25$
- À distance 2 : $\binom{5}{2} \times 5^2 = 250$



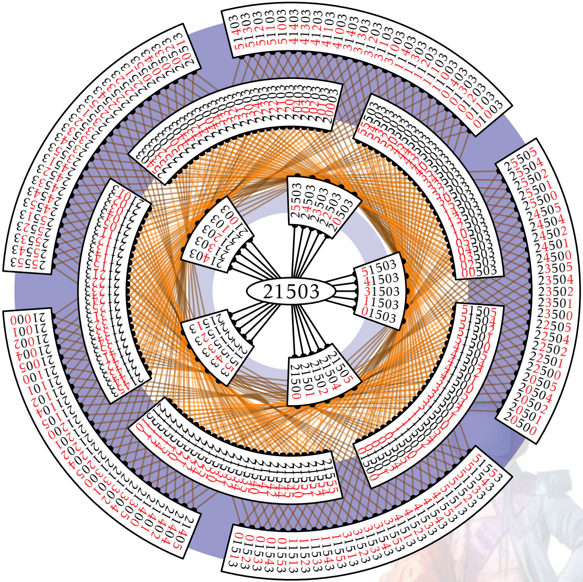
Boule de Hamming

- Alphabet de taille 6.
- Mots de longueur 5.
- À distance 1 : $\binom{5}{1} \times 5^1 = 25$
- À distance 2 : $\binom{5}{2} \times 5^2 = 250$
- Volume de la boule de rayon 2 :



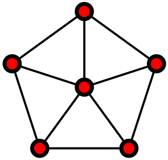
Boule de Hamming

- Alphabet de taille 6.
- Mots de longueur 5.
- À distance 1 : $\binom{5}{1} \times 5^1 = 25$
- À distance 2 : $\binom{5}{2} \times 5^2 = 250$
- Volume de la boule de rayon 2 : $1 + 25 + 250 = 276$



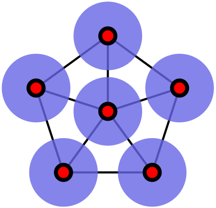
Borne de Hamming

- Pour corriger t erreurs, il faut des boules disjointes de rayon t autour de chaque mot du code.



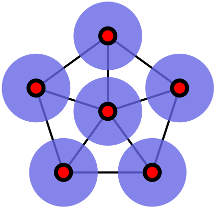
Borne de Hamming

- Pour corriger t erreurs, il faut des boules disjointes de rayon t autour de chaque mot du code.



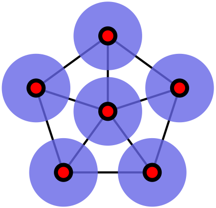
Borne de Hamming

- Pour corriger t erreurs, il faut des boules disjointes de rayon t autour de chaque mot du code.
- q : taille de l'alphabet
- n : nombre de symboles d'un mot du code
- M : nombre de mots du code

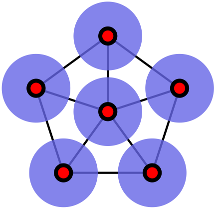


Borne de Hamming

- Pour corriger t erreurs, il faut des boules disjointes de rayon t autour de chaque mot du code.
- q : taille de l'alphabet
- n : nombre de symboles d'un mot du code
- M : nombre de mots du code
- Volume d'une boule de rayon t : $V_t = \sum_{i=0}^{t} \binom{n}{i} (q-1)^i$

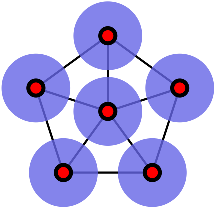


Borne de Hamming



- Pour corriger t erreurs, il faut des boules disjointes de rayon t autour de chaque mot du code.
- q : taille de l'alphabet
- n : nombre de symboles d'un mot du code
- M : nombre de mots du code
- Volume d'une boule de rayon t : $\mathrm{V}_t = \sum_{i=0}^{t} \binom{n}{i} (q-1)^i$
- Il faut donc avoir : $\mathrm{M} \times \mathrm{V}_t \leq q^n$

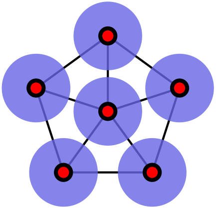
Borne de Hamming



- Pour corriger t erreurs, il faut des boules disjointes de rayon t autour de chaque mot du code.
- q : taille de l'alphabet
- n : nombre de symboles d'un mot du code
- M : nombre de mots du code
- Volume d'une boule de rayon t : $V_t = \sum_{i=0}^{t} \binom{n}{i}(q-1)^i$
- Il faut donc avoir : $M \times V_t \leq q^n$
- Dans notre cas, on veut le plus petit $n \in \mathbb{N}$ tel que :

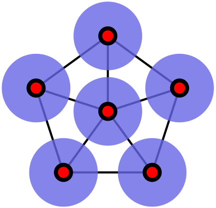
$2^{25} \times V_1 \leq 2^n$, avec $V_1 = \binom{n}{0} + \binom{n}{1} = 1 + n$

Borne de Hamming



- Pour corriger t erreurs, il faut des boules disjointes de rayon t autour de chaque mot du code.
- q : taille de l'alphabet
- n : nombre de symboles d'un mot du code
- M : nombre de mots du code
- Volume d'une boule de rayon t : $V_t = \sum_{i=0}^{t} \binom{n}{i}(q-1)^i$
- Il faut donc avoir : $M \times V_t \leq q^n$
- Dans notre cas, on veut le plus petit $n \in \mathbb{N}$ tel que :
  $2^{25} \times V_1 \leq 2^n$, avec $V_1 = \binom{n}{0} + \binom{n}{1} = 1 + n$
- La plus petite valeur entière est 30.

Borne de Hamming



- Pour corriger t erreurs, il faut des boules disjointes de rayon t autour de chaque mot du code.
- q : taille de l'alphabet
- n : nombre de symboles d'un mot du code
- M : nombre de mots du code
- Volume d'une boule de rayon t : $V_t = \sum_{i=0}^{t} \binom{n}{i} (q-1)^i$
- Il faut donc avoir : $M \times V_t \leq q^n$
- Dans notre cas, on veut le plus petit $n \in \mathbb{N}$ tel que :

  $2^{25} \times V_1 \leq 2^n$, avec $V_1 = \binom{n}{0} + \binom{n}{1} = 1 + n$
- La plus petite valeur entière est 30.
- Il faut rajouter au moins 5 bits.

Un code avec 5 bits de parité

- On fait un tableau indiquant quels bits du message sont utilisés pour quels bits de parité.
- Chaque bit du message est utilisé dans 2, 3 ou 4 bits de parité.

bits du message	1 à 10		11 à 20		21 à 25
1[er] bit de parité					
2[e] bit de parité					
3[e] bit de parité					
4[e] bit de parité					
5[e] bit de parité					

Un code avec 5 bits de parité

- On fait un tableau indiquant quels bits du message sont utilisés pour quels bits de parité.
- Chaque bit du message est utilisé dans 2, 3 ou 4 bits de parité.

bits du message	1 à 10										11 à 20		21 à 25
1^er^ bit de parité	1	1	1	1	0	0	0	0	0	0			
2^e^ bit de parité	1	0	0	0	1	1	1	0	0	0			
3^e^ bit de parité	0	1	0	0	1	0	0	1	1	0			
4^e^ bit de parité	0	0	1	0	0	1	0	1	0	1			
5^e^ bit de parité	0	0	0	1	0	0	1	0	1	1			

Un code avec 5 bits de parité

- On fait un tableau indiquant quels bits du message sont utilisés pour quels bits de parité.
- Chaque bit du message est utilisé dans 2, 3 ou 4 bits de parité.

bits du message	1 à 10										11 à 20										21 à 25
1^{er} bit de parité	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	
2^{e} bit de parité	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	
3^{e} bit de parité	0	1	0	0	1	0	0	1	1	0	1	0	0	1	1	0	1	1	0	1	
4^{e} bit de parité	0	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	1	0	1	1	
5^{e} bit de parité	0	0	0	1	0	0	1	0	1	1	0	0	1	0	1	1	0	1	1	1	

Un code avec 5 bits de parité

- On fait un tableau indiquant quels bits du message sont utilisés pour quels bits de parité.
- Chaque bit du message est utilisé dans 2, 3 ou 4 bits de parité.

bits du message	1 à 10										11 à 20										21 à 25				
1^er^ bit de parité	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	0
2^e^ bit de parité	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	1	1	1	0	1
3^e^ bit de parité	0	1	0	0	1	0	0	1	1	0	1	0	0	1	1	0	1	1	0	1	1	1	0	1	1
4^e^ bit de parité	0	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	1	0	1	1	1	0	1	1	1
5^e^ bit de parité	0	0	0	1	0	0	1	0	1	1	0	0	1	0	1	1	0	1	1	1	0	1	1	1	1

Un code avec 5 bits de parité

- On fait un tableau indiquant quels bits du message sont utilisés pour quels bits de parité.
- Chaque bit du message est utilisé dans 2, 3 ou 4 bits de parité.

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	1 1 1 1 0	0 0 0 0 0	1 1 1 1 1	1 0 0 0 0	1 1 1 1 0
2^{e} bit de parité	1 0 0 0 1	1 1 0 0 0	1 1 1 0 0	0 1 1 1 0	1 1 1 0 1
3^{e} bit de parité	0 1 0 0 1	0 0 1 1 0	1 0 0 1 1	0 1 1 0 1	1 1 0 1 1
4^{e} bit de parité	0 0 1 0 0	1 0 1 0 1	0 1 0 1 0	1 1 0 1 1	1 0 1 1 1
5^{e} bit de parité	0 0 0 1 0	0 1 0 1 1	0 0 1 0 1	1 0 1 1 1	0 1 1 1 1

- Il pourrait y avoir un 26^{e} bit dans le message.

Principe de vérification

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	11110	00000	11111	10000	11110
2e bit de parité	10001	11000	11100	01110	11101
3e bit de parité	01001	00110	10011	01101	11011
4e bit de parité	00100	10101	01010	11011	10111
5e bit de parité	00010	01011	00101	10111	01111

- À la réception, on recalcule les bits de parité :

Principe de vérification

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	11110	00000	11111	10000	11110
2e bit de parité	10001	11000	11100	01110	11101
3e bit de parité	01001	00110	10011	01101	11011
4e bit de parité	00100	10101	01010	11011	10111
5e bit de parité	00010	01011	00101	10111	01111

- À la réception, on recalcule les bits de parité :
  - Pas de différence : tout va bien

Principe de vérification

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	11110	00000	11111	10000	11110
2e bit de parité	10001	11000	11100	01110	11101
3e bit de parité	01001	00110	10011	01101	11011
4e bit de parité	00100	10101	01010	11011	10111
5e bit de parité	00010	01011	00101	10111	01111

- À la réception, on recalcule les bits de parité :
  - Pas de différence : tout va bien
  - 1 bit différent : un bit de parité a été altéré.

Principe de vérification

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	11110	00000	11111	10000	11110
2e bit de parité	10001	11000	11100	01110	11101
3e bit de parité	01001	00110	10011	01101	11011
4e bit de parité	00100	10101	01010	11011	10111
5e bit de parité	00010	01011	00101	10111	01111

- À la réception, on recalcule les bits de parité :
  - Pas de différence : tout va bien
  - 1 bit différent : un bit de parité a été altéré.
  - 2, 3 ou 4 bits différents : un des bits du message a changé.

Principe de vérification

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	11110	00000	11111	10000	11110
2e bit de parité	10001	11000	11100	01110	11101
3e bit de parité	01001	00110	10011	01101	11011
4e bit de parité	00100	10101	01010	11011	10111
5e bit de parité	00010	01011	00101	10111	01111

- À la réception, on recalcule les bits de parité :
  - Pas de différence : tout va bien
  - 1 bit différent : un bit de parité a été altéré.
  - 2, 3 ou 4 bits différents : un des bits du message a changé.
  - 5 bits différents : impossible sous l'hypothèse qu'il n'y a qu'une erreur au plus.

Principe de vérification

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	11110	00000	11111	10000	11110
2^{e} bit de parité	10001	11000	11100	01110	11101
3^{e} bit de parité	01001	00110	10011	01101	11011
4^{e} bit de parité	00100	10101	01010	11011	10111
5^{e} bit de parité	00010	01011	00101	10111	01111

- À la réception, on recalcule les bits de parité :
  - Pas de différence : tout va bien
  - 1 bit différent : un bit de parité a été altéré.
  - 2, 3 ou 4 bits différents : un des bits du message a changé.
  - 5 bits différents : impossible sous l'hypothèse qu'il n'y a qu'une erreur au plus.
- Les bits différents indiquent l'erreur.

Principe de vérification

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	11110	00000	11111	10000	11110
2e bit de parité	10001	11000	11100	01110	11101
3e bit de parité	01001	00110	10011	01101	11011
4e bit de parité	00100	10101	01010	11011	10111
5e bit de parité	00010	01011	00101	10111	01111

- À la réception, on recalcule les bits de parité :
  - Pas de différence : tout va bien
  - 1 bit différent : un bit de parité a été altéré.
  - 2, 3 ou 4 bits différents : un des bits du message a changé.
  - 5 bits différents : impossible sous l'hypothèse qu'il n'y a qu'une erreur au plus.
- Les bits différents indiquent l'erreur.
- Par exemple, si ce sont les bits de parité 1, 3 et 4 qui sont différents, c'est le

Principe de vérification

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	11110	00000	11111	10000	11110
2e bit de parité	10001	11000	11100	01110	11101
3e bit de parité	01001	00110	10011	01101	11011
4e bit de parité	00100	10101	01010	11011	10111
5e bit de parité	00010	01011	00101	10111	01111

- À la réception, on recalcule les bits de parité :
  - Pas de différence : tout va bien
  - 1 bit différent : un bit de parité a été altéré.
  - 2, 3 ou 4 bits différents : un des bits du message a changé.
  - 5 bits différents : impossible sous l'hypothèse qu'il n'y a qu'une erreur au plus.
- Les bits différents indiquent l'erreur.
- Par exemple, si ce sont les bits de parité 1, 3 et 4 qui sont différents, c'est le 14e bit du message qui a été modifié.

Un exemple plus concret

- On reçoit $\overbrace{10000\ 10110\ 11111\ 00100\ 01010}^{\text{message}}\ \color{red}{\overbrace{01111}^{\text{parité}}}$.

Un exemple plus concret

- On reçoit $\overbrace{\text{10000 10110 11111 00100 01010}}^{\text{message}}$ $\color{red}{\overbrace{\text{01111}}^{\text{parité}}}$.
- On recalcule les bits de parité :

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	11110	00000	11111	10000	11110
2e bit de parité	10001	11000	11100	01110	11101
3e bit de parité	01001	00110	10011	01101	11011
4e bit de parité	00100	10101	01010	11011	10111
5e bit de parité	00010	01011	00101	10111	01111

Un exemple plus concret

- On reçoit $\overbrace{10000\ 10110\ 11111\ 00100\ 01010}^{\text{message}}\ \overbrace{01111}^{\text{parité}}$.
- On recalcule les bits de parité :

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	11110	00000	11111	10000	11110
2^{e} bit de parité	10001	11000	11100	01110	11101
3^{e} bit de parité	01001	00110	10011	01101	11011
4^{e} bit de parité	00100	10101	01010	11011	10111
5^{e} bit de parité	00010	01011	00101	10111	01111

Un exemple plus concret

- On reçoit $\overbrace{10000\ 10110\ 11111\ 00100\ 01010}^{\text{message}}\ \overbrace{01111}^{\text{parité}}$.
- On recalcule les bits de parité : 0

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	11110	00000	11111	10000	11110
2e bit de parité	10001	11000	11100	01110	11101
3e bit de parité	01001	00110	10011	01101	11011
4e bit de parité	00100	10101	01010	11011	10111
5e bit de parité	00010	01011	00101	10111	01111

Un exemple plus concret

- On reçoit $\underbrace{10000\ 10110\ 11111\ 00100\ 01010}_{\text{message}}\ \underbrace{01111}_{\text{parité}}$.
- On recalcule les bits de parité : 01

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	11110	00000	11111	10000	11110
2e bit de parité	10001	11000	11100	01110	11101
3e bit de parité	01001	00110	10011	01101	11011
4e bit de parité	00100	10101	01010	11011	10111
5e bit de parité	00010	01011	00101	10111	01111

Un exemple plus concret

- On reçoit $\overbrace{10000\ 10110\ 11111\ 00100\ 01010}^{\text{message}}\ \overbrace{01111}^{\text{parité}}$.
- On recalcule les bits de parité : 010

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	11110	00000	11111	10000	11110
2e bit de parité	10001	11000	11100	01110	11101
3e bit de parité	01001	00110	10011	01101	11011
4e bit de parité	00100	10101	01010	11011	10111
5e bit de parité	00010	01011	00101	10111	01111

Un exemple plus concret

- On reçoit $\overbrace{10000\ 10110\ 11111\ 00100\ 01010}^{\text{message}}\ \overbrace{01111}^{\text{parité}}$.
- On recalcule les bits de parité : 0101

bits du message	1 à 10		11 à 20		21 à 25
1^er^ bit de parité	11110	00000	11111	10000	11110
2^e^ bit de parité	10001	11000	11100	01110	11101
3^e^ bit de parité	01001	00110	10011	01101	11011
4^e^ bit de parité	00100	10101	01010	11011	10111
5^e^ bit de parité	00010	01011	00101	10111	01111

Un exemple plus concret

- On reçoit $\overbrace{10000\ 10110\ 11111\ 00100\ 01010}^{\text{message}}\ \color{red}{\overbrace{01111}^{\text{parité}}}$.
- On recalcule les bits de parité : 01010

bits du message	1 à 10		11 à 20		21 à 25
1^{er} bit de parité	11110	00000	11111	10000	11110
2^{e} bit de parité	10001	11000	11100	01110	11101
3^{e} bit de parité	01001	00110	10011	01101	11011
4^{e} bit de parité	00100	10101	01010	11011	10111
5^{e} bit de parité	00010	01011	00101	10111	01111

Un exemple plus concret

- On reçoit $\overbrace{10000\ 10110\ 11111\ 00100\ 01010}^{\text{message}}\ \color{red}{\overbrace{01111}^{\text{parité}}}$.
- On recalcule les bits de parité : 01010
- On cherche les différences : $01111 \oplus 01010 =$

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	11110	00000	11111	10000	11110
2e bit de parité	10001	11000	11100	01110	11101
3e bit de parité	01001	00110	10011	01101	11011
4e bit de parité	00100	10101	01010	11011	10111
5e bit de parité	00010	01011	00101	10111	01111

Un exemple plus concret

- On reçoit $\overbrace{10000\ 10110\ 11111\ 00100\ 01010}^{\text{message}}\ \overbrace{01111}^{\text{parité}}$.
- On recalcule les bits de parité : 01010
- On cherche les différences : $01111 \oplus 01010 = 00101$

bits du message	1 à 10		11 à 20		21 à 25
1$^{\text{er}}$ bit de parité	11110	00000	11111	10000	11110
2$^{\text{e}}$ bit de parité	10001	11000	11100	01110	11101
3$^{\text{e}}$ bit de parité	01001	00110	10011	01101	11011
4$^{\text{e}}$ bit de parité	00100	10101	01010	11011	10111
5$^{\text{e}}$ bit de parité	00010	01011	00101	10111	01111

Un exemple plus concret

- On reçoit $\overbrace{10000\ 10110\ 11111\ 00100\ 01010}^{\text{message}}\ \overbrace{01111}^{\text{parité}}$.
- On recalcule les bits de parité : 01010
- On cherche les différences : $01111 \oplus 01010 = 00101$
- C'est donc le bit du message qui a changé.

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	11110	00000	11111	10000	11110
2e bit de parité	10001	11000	11100	01110	11101
3e bit de parité	01001	00110	10011	01101	11011
4e bit de parité	00100	10101	01010	11011	10111
5e bit de parité	00010	01011	00101	10111	01111

Un exemple plus concret

- On reçoit $\overbrace{10000\ 10110\ 11111\ 00100\ 01010}^{\text{message}}\ \overbrace{01111}^{\text{parité}}$.
- On recalcule les bits de parité : 01010
- On cherche les différences : $01111 \oplus 01010 = 00101$
- C'est donc le 9^{e} bit du message qui a changé.

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	11110	00000	11111	10000	11110
2^{e} bit de parité	10001	11000	11100	01110	11101
3^{e} bit de parité	01001	00110	10011	01101	11011
4^{e} bit de parité	00100	10101	01010	11011	10111
5^{e} bit de parité	00010	01011	00101	10111	01111

Un exemple plus concret

- On reçoit $\overbrace{10000\ 10110\ 11111\ 00100\ 01010}^{\text{message}}\ \overbrace{01111}^{\text{parité}}$.
- On recalcule les bits de parité : 01010
- On cherche les différences : $01111 \oplus 01010 = 00101$
- C'est donc le 9^{e} bit du message qui a changé.
- Le message original était donc :

$$10000\ 10100\ 11111\ 00100\ 01010\ 01111$$

bits du message	1 à 10		11 à 20		21 à 25
1er bit de parité	11110	00000	11111	10000	11110
2^{e} bit de parité	10001	11000	11100	01110	11101
3^{e} bit de parité	01001	00110	10011	01101	11011
4^{e} bit de parité	00100	10101	01010	11011	10111
5^{e} bit de parité	00010	01011	00101	10111	01111

Codage de l'information

- Tout est représenté en binaire dans un ordinateur.

Codage de l’information

- Tout est représenté en binaire dans un ordinateur.
- En pratique, on regroupe les bits par 8. Cela donne un **octet**.

Codage de l'information

- Tout est représenté en binaire dans un ordinateur.
- En pratique, on regroupe les bits par 8. Cela donne un **octet**.
- On les affiche avec 2 chiffres en **hexadécimal** (de 0 à F).

00000000	25	50	44	46	2D	31	2E	35	0A	25	D0	D4	C5	D8	0A	31	30	20	%PDF-1.5.%.....10
00000012	30	20	6F	62	6A	0A	3C	3C	0A	2F	54	79	70	65	20	2F	58	4F	0 obj.<<./Type /XO
00000024	62	6A	65	63	74	0A	2F	53	75	62	74	79	70	65	20	2F	46	6F	bject./Subtype /Fo
00000036	72	6D	0A	2F	42	42	6F	78	20	5B	30	20	30	20	31	30	30	20	rm./BBox [0 0 100
00000048	31	30	30	5D	0A	2F	46	6F	72	6D	54	79	70	65	20	31	0A	2F	100]./FormType 1./
0000005a	4D	61	74	72	69	78	20	5B	31	20	30	20	30	20	31	20	30	20	Matrix [1 0 0 1 0
0000006c	30	5D	0A	2F	52	65	73	6F	75	72	63	65	73	20	31	31	20	30	0]./Resources 11 0
0000007e	20	52	0A	2F	4C	65	6E	67	74	68	20	31	35	20	20	20	20	20	R./Length 15
00000090	20	20	20	0A	2F	46	69	6C	74	65	72	20	2F	46	6C	61	74	65	./Filter /Flate
000000a2	44	65	63	6F	64	65	0A	3E	3E	0A	73	74	72	65	61	6D	0A	78	Decode.>>.stream.x
000000b4	DA	D3	0F	CE	50	28	CE	E0	02	00	07	FD	01	F0	0A	65	6E	64	....P(.........end
000000c6	73	74	72	65	61	6D	0A	65	6E	64	6F	62	6A	0A	31	33	20	30	stream.endobj.13 0
000000d8	20	6F	62	6A	0A	3C	3C	0A	2F	54	79	70	65	20	2F	58	4F	62	obj.<<./Type /XOb
000000ea	6A	65	63	74	0A	2F	53	75	62	74	79	70	65	20	2F	46	6F	72	ject./Subtype /For
000000cf	6D	0A	2F	42	42	6F	78	20	5B	30	20	30	20	31	30	30	20	31	m./BBox [0 0 100 1
0000010e	30	30	5D	0A	2F	46	6F	72	6D	54	79	70	65	20	31	0A	2F	4D	00]./FormType 1./M
00000120	61	74	72	69	78	20	5B	31	20	30	20	30	20	31	20	30	20	30	atrix [1 0 0 1 0 0
00000132	5D	0A	2F	52	65	73	6F	75	72	63	65	73	20	31	34	20	30	20	]./Resources 14 0
00000144	52	0A	2F	4C	65	6E	67	74	68	20	31	35	20	20	20	20	20	20	R./Length 15
00000156	20	20	0A	2F	46	69	6C	74	65	72	20	2F	46	6C	61	74	65	44	./Filter /FlateD
00000168	65	63	6F	64	65	0A	3E	3E	0A	73	74	72	65	61	6D	0A	78	DA	ecode.>>.stream.x.

Codage de l'information

- Tout est représenté en binaire dans un ordinateur.
- En pratique, on regroupe les bits par 8. Cela donne un **octet**.
- On les affiche avec 2 chiffres en **hexadécimal** (de 0 à F).

```
00000000 25 50 44 46 2D 31 2E 35 0A 25 D0 D4 C5 D8 0A 31 30 20 %PDF-1.5.%.....10
00000012 30 20 6F 62 6A 0A 3C 3C 0A 2F 54 79 70 65 20 2F 58 4F 0 obj.<<./Type /XO
00000024 62 6A 65 63 74 0A 2F 53 75 62 74 79 70 65 20 2F 46 6F bject./Subtype /Fo
00000036 72 6D 0A 2F 42 42 6F 78 20 5B 30 20 30 20 31 30 30 20 rm./BBox [0 0 100
00000048 31 30 30 5D 0A 2F 46 6F 72 6D 54 79 70 65 20 31 0A 2F 100]./FormType 1./
0000005a 4D 61 74 72 69 78 20 5B 31 20 30 20 30 20 31 20 30 20 Matrix [1 0 0 1 0
0000006c 30 5D 0A 2F 52 65 73 6F 75 72 63 65 73 20 31 31 20 30 0]./Resources 11 0
0000007e 20 52 0A 2F 4C 65 6E 67 74 68 20 31 35 20 20 20 20 20  R./Length 15
00000090 20 20 20 0A 2F 46 69 6C 74 65 72 20 2F 46 6C 61 74 65    ./Filter /Flate
000000a2 44 65 63 6F 64 65 0A 3E 3E 0A 73 74 72 65 61 6D 0A 78 Decode.>>.stream.x
000000b4 DA D3 0F CE 50 28 CE E0 02 00 07 FD 01 F0 0A 65 6E 64 ....P(.........end
000000c6 73 74 72 65 61 6D 0A 65 6E 64 6F 62 6A 0A 31 33 20 30 stream.endobj.13 0
000000d8 20 6F 62 6A 0A 3C 3C 0A 2F 54 79 70 65 20 2F 58 4F 62  obj.<<./Type /XOb
000000ea 6A 65 63 74 0A 2F 53 75 62 74 79 70 65 20 2F 46 6F 72 ject./Subtype /For
000000cf 6D 0A 2F 42 42 6F 78 20 5B 30 20 30 20 31 30 30 20 31 m./BBox [0 0 100 1
0000010e 30 30 5D 0A 2F 46 6F 72 6D 54 79 70 65 20 31 0A 2F 4D 00]./FormType 1./M
00000120 61 74 72 69 78 20 5B 31 20 30 20 30 20 31 20 30 20 30 atrix [1 0 0 1 0 0
00000132 5D 0A 2F 52 65 73 6F 75 72 63 65 73 20 31 34 20 30 20 ]./Resources 14 0
00000144 52 0A 2F 4C 65 6E 67 74 68 20 31 35 20 20 20 20 20 20 R./Length 15
00000156 20 20 0A 2F 46 69 6C 74 65 72 20 2F 46 6C 61 74 65 44   ./Filter /FlateD
00000168 65 63 6F 64 65 0A 3E 3E 0A 73 74 72 65 61 6D 0A 78 DA ecode.>>.stream.x.
```

- $2^8 = 16^2 = 256$

Codage des caractères

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Nombres décimaux

- Décimaux en binaire :

Binaire		Décimal
101,11	=	

Nombres décimaux

- Décimaux en binaire :

Binaire		Décimal
101,11	$=$	$1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}$

Nombres décimaux

- Décimaux en binaire :

Binaire		Décimal
101,11	$=$	$1\times2^{2}+0\times2^{1}+1\times2^{0}+1\times2^{-1}+1\times2^{-2}$
	$=$	$4+1+0{,}5+0{,}25$

Nombres décimaux

- Décimaux en binaire :

Binaire		Décimal
101,11	$=$	$1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}$
	$=$	$4+1+0,5+0,25$
	$=$	$5,75$

Nombres décimaux

- Décimaux en binaire :

$$\begin{array}{ccl} \text{Binaire} & & \text{Décimal} \\ 101{,}11 & = & 1\times 2^2 + 0\times 2^1 + 1\times 2^0 + 1\times 2^{-1} + 1\times 2^{-2} \\ & = & 4 + 1 + 0{,}5 + 0{,}25 \\ & = & 5{,}75 \end{array}$$

- Convertir la partie décimale en binaire :

$0{,}3 \rightarrow 0{,}$

Nombres décimaux

- Décimaux en binaire :

Binaire		Décimal
101,11	$=$	$1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}$
	$=$	$4+1+0{,}5+0{,}25$
	$=$	$5{,}75$

- Convertir la partie décimale en binaire :

$0{,}3\times2=$

$0{,}3\to0{,}$

Nombres décimaux

- Décimaux en binaire :

$$\begin{array}{rcl} \text{Binaire} & & \text{Décimal} \\ 101{,}11 & = & 1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2} \\ & = & 4+1+0{,}5+0{,}25 \\ & = & 5{,}75 \end{array}$$

- Convertir la partie décimale en binaire :

$$0,3\times2=0,6$$

$0{,}3 \rightarrow 0,$

Nombres décimaux

- Décimaux en binaire :

Binaire		Décimal
101,11	$=$	$1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}$
	$=$	$4+1+0,5+0,25$
	$=$	$5,75$

- Convertir la partie décimale en binaire :

$$0,3\times 2=\boxed{0},6$$

$$0,3\rightarrow 0,0$$

Nombres décimaux

- Décimaux en binaire :

Binaire		Décimal
101,11	$=$	$1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}$
	$=$	$4+1+0,5+0,25$
	$=$	$5,75$

- Convertir la partie décimale en binaire :

$$0,3\times2=\boxed{0},6$$

$$0,3\rightarrow0,0$$

Nombres décimaux

- Décimaux en binaire :

$$\begin{array}{ccl} \text{Binaire} & & \text{Décimal} \\ 101{,}11 & = & 1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2} \\ & = & 4+1+0{,}5+0{,}25 \\ & = & 5{,}75 \end{array}$$

- Convertir la partie décimale en binaire :

$$0{,}3\times2=\boxed{0}{,}6$$
$$0{,}6\times2=$$

$0{,}3 \rightarrow 0{,}0$

Nombres décimaux

- Décimaux en binaire :

$$\begin{array}{ccl} \text{Binaire} & & \text{Décimal} \\ 101{,}11 & = & 1\times2^2 + 0\times2^1 + 1\times2^0 + 1\times2^{-1} + 1\times2^{-2} \\ & = & 4 + 1 + 0{,}5 + 0{,}25 \\ & = & 5{,}75 \end{array}$$

- Convertir la partie décimale en binaire :

$$0,3 \times 2 = \boxed{0},6$$
$$0,6 \times 2 = 1,2$$

$$0,3 \rightarrow 0,0$$

Nombres décimaux

- Décimaux en binaire :

Binaire		Décimal
101,11	$=$	$1\times 2^2 + 0\times 2^1 + 1\times 2^0 + 1\times 2^{-1} + 1\times 2^{-2}$
	$=$	$4 + 1 + 0{,}5 + 0{,}25$
	$=$	$5{,}75$

- Convertir la partie décimale en binaire :

$$0{,}3 \times 2 = \boxed{0}{,}6$$
$$0{,}6 \times 2 = \boxed{1}{,}2$$

$0{,}3 \rightarrow 0{,}0\ 1$

Nombres décimaux

- Décimaux en binaire :

Binaire		Décimal
101,11	$=$	$1\times2^{2}+0\times2^{1}+1\times2^{0}+1\times2^{-1}+1\times2^{-2}$
	$=$	$4+1+0{,}5+0{,}25$
	$=$	$5{,}75$

- Convertir la partie décimale en binaire :

$$0{,}3\times2=\boxed{0}{,}6$$
$$0{,}6\times2=\boxed{1}{,}2$$

$0{,}3 \rightarrow 0{,}0\ 1$

Nombres décimaux

- Décimaux en binaire :

$$\begin{array}{rcl} \text{Binaire} & & \text{Décimal} \\ 101{,}11 & = & 1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2} \\ & = & 4+1+0{,}5+0{,}25 \\ & = & 5{,}75 \end{array}$$

- Convertir la partie décimale en binaire :

$$\begin{aligned} 0{,}3\times2 &= \boxed{0}{,}6 \\ 0{,}6\times2 &= \boxed{1}{,}2 \\ 0{,}2\times2 &= \boxed{0}{,}4 \\ 0{,}4\times2 &= \boxed{0}{,}8 \\ 0{,}8\times2 &= \boxed{1}{,}6 \end{aligned}$$

$$0{,}3 \rightarrow 0{,}0\ 1001$$

Nombres décimaux

- Décimaux en binaire :

Binaire		Décimal
101,11	$=$	$1\times 2^2+0\times 2^1+1\times 2^0+1\times 2^{-1}+1\times 2^{-2}$
	$=$	$4+1+0{,}5+0{,}25$
	$=$	$5{,}75$

- Convertir la partie décimale en binaire :

$$0{,}3\times 2=\boxed{0},6$$
$$0{,}6\times 2=\boxed{1},2$$
$$0{,}2\times 2=\boxed{0},4$$
$$0{,}4\times 2=\boxed{0},8$$
$$0{,}8\times 2=\boxed{1},6$$

$$0{,}3\longrightarrow 0{,}0\ 1001$$

Nombres décimaux

- Décimaux en binaire :

$$\begin{array}{ccl} \text{Binaire} & & \text{Décimal} \\ 101{,}11 & = & 1\times 2^2 + 0\times 2^1 + 1\times 2^0 + 1\times 2^{-1} + 1\times 2^{-2} \\ & = & 4 + 1 + 0{,}5 + 0{,}25 \\ & = & 5{,}75 \end{array}$$

- Convertir la partie décimale en binaire :

$$\begin{aligned} 0{,}3 \times 2 &= \boxed{0},6 \\ 0{,}6 \times 2 &= \boxed{1},2 \\ 0{,}2 \times 2 &= \boxed{0},4 \\ 0{,}4 \times 2 &= \boxed{0},8 \\ 0{,}8 \times 2 &= \boxed{1},6 \end{aligned}$$

$$0{,}3 \rightarrow 0{,}0[1001]$$

Nombres décimaux

- Décimaux en binaire :

$$\begin{array}{ccl} \text{Binaire} & & \text{Décimal} \\ 101{,}11 & = & 1\times 2^{2} + 0\times 2^{1} + 1\times 2^{0} + 1\times 2^{-1} + 1\times 2^{-2} \\ & = & 4 + 1 + 0{,}5 + 0{,}25 \\ & = & 5{,}75 \end{array}$$

- Convertir la partie décimale en binaire :

$$\begin{array}{l} 0,3\times 2 = \boxed{0},6 \\ 0,6\times 2 = \boxed{1},2 \\ 0,2\times 2 = \boxed{0},4 \\ 0,4\times 2 = \boxed{0},8 \\ 0,8\times 2 = \boxed{1},6 \end{array} \qquad \begin{array}{l} 0,2\times 2 = \boxed{0},4 \\ 0,4\times 2 = \boxed{0},8 \\ 0,8\times 2 = \boxed{1},6 \\ 0,6\times 2 = \boxed{1},2 \end{array}$$

$0,3 \rightarrow 0,0[1001]$ $\qquad$ $0,2 \rightarrow 0,[0011]$

Nombres décimaux

- Décimaux en binaire :

Binaire		Décimal
101,11	$=$	$1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}$
	$=$	$4+1+0{,}5+0{,}25$
	$=$	$5{,}75$

- Convertir la partie décimale en binaire :

$$
\begin{aligned}
0{,}3\times2&=\boxed{0},6\\
0{,}6\times2&=\boxed{1},2\\
0{,}2\times2&=\boxed{0},4\\
0{,}4\times2&=\boxed{0},8\\
0{,}8\times2&=\boxed{1},6
\end{aligned}
\qquad
\begin{aligned}
0{,}2\times2&=\boxed{0},4\\
0{,}4\times2&=\boxed{0},8\\
0{,}8\times2&=\boxed{1},6\\
0{,}6\times2&=\boxed{1},2
\end{aligned}
\qquad
\begin{aligned}
0{,}5\times2&=\boxed{1},0\\
0{,}0\times2&=\boxed{0},0
\end{aligned}
$$

$0{,}3\longrightarrow0{,}0[1001]$ $\qquad$ $0{,}2\longrightarrow0{,}[0011]$ $\qquad$ $0{,}5\longrightarrow0{,}1$

Virgule flottante

- Notation scientifique :

Décimal		Binaire
211	=	11010011
0,05	=	

Virgule flottante

- Notation scientifique :

Décimal		Binaire
211	$=$	11010011
$2{,}11 \times 10^2$	$=$	
$0{,}05$	$=$	

Virgule flottante

- Notation scientifique :

Décimal		Binaire
211	$=$	11010011
$2{,}11 \times 10^2$	$=$	$1{,}1010011 \times 2^7$
$0{,}05$	$=$	

Virgule flottante

- Notation scientifique :

Décimal		Binaire
211	$=$	11010011
$2{,}11 \times 10^2$	$=$	$1{,}1010011 \times 2^7$
$0{,}05$	$=$	$0{,}000011[0011]$

Virgule flottante

- Notation scientifique :

Décimal		Binaire
211	$=$	11010011
$2,11 \times 10^2$	$=$	$1,1010011 \times 2^7$
$0,05$	$=$	$0,000011[0011]$
5×10^{-2}	$=$	

Virgule flottante

- Notation scientifique :

Décimal		Binaire
211	$=$	11010011
$2{,}11 \times 10^{2}$	$=$	$1{,}1010011 \times 2^{7}$
$0{,}05$	$=$	$0{,}000011[0011]$
5×10^{-2}	$=$	$1{,}1[0011] \times 2^{-5}$

Virgule flottante

- Notation scientifique :

Décimal		Binaire
211	$=$	11010011
$2{,}11 \times 10^2$	$=$	$1{,}1010011 \times 2^7$
$0{,}05$	$=$	$0{,}000011[0011]$
5×10^{-2}	$=$	$1{,}1[0011] \times 2^{-5}$

- Représentation en virgule flottante sur 32 bits :

$$s\,\underbrace{eeeeeeee}_{\text{8 bits}}\,\underbrace{mmmmmmmmmmmmmmmmmmmmmmm}_{\text{23 bits}}$$

Virgule flottante

- Notation scientifique :

Décimal		Binaire
211	$=$	11010011
$2{,}11 \times 10^{2}$	$=$	$1{,}1010011 \times 2^{7}$
$0{,}05$	$=$	$0{,}000011[0011]$
5×10^{-2}	$=$	$1{,}1[0011] \times 2^{-5}$

- Représentation en virgule flottante sur 32 bits :

$$s\,\overbrace{\underbrace{eeeeeeee}_{8\text{ bits}}}^{\text{exposant}}\,\overbrace{\underbrace{mmmmmmmmmmmmmmmmmmmmmmm}_{23\text{ bits}}}^{\text{mantisse}}$$

- $\text{Nombre} = (-1)^{s} \times 1{,}\text{mantisse} \times 2^{\text{exposant}-127}$

Virgule flottante (suite)

- Écriture de 211,3

Décimal	Binaire
211,3	

On obtient :

Virgule flottante (suite)

- Écriture de 211,3

Décimal		Binaire
211,3	$\approx$	11010011,0100110011001101

On obtient :

Virgule flottante (suite)

- Écriture de 211,3

Décimal		Binaire
211,3	$\approx$	11010011,0100110011001101
	$\approx$	$1{,}10100110100110011001101 \times 2^7$

On obtient :

Virgule flottante (suite)

- Écriture de 211,3

Décimal		Binaire
211,3	$\approx$	11010011,0100110011001101
	$\approx$	$1,10100110100110011001101 \times 2^7$

On obtient : 10100110100110011001101

Virgule flottante (suite)

- Écriture de 211,3

Décimal		Binaire
$211{,}3$	$\approx$	$11010011{,}0100110011001101$
	$\approx$	$1{,}10100110100110011001101 \times 2^7$
$7+127$	$=$	

On obtient : 10100110100110011001101

Virgule flottante (suite)

- Écriture de 211,3

Décimal		Binaire
211,3	$\approx$	11010011,0100110011001101
	$\approx$	$1{,}10100110100110011001101 \times 2^7$
$7+127$	$=$	10000110

On obtient : 10100110100110011001101

Virgule flottante (suite)

- Écriture de 211,3

Décimal		Binaire
211,3	$\approx$	11010011,0100110011001101
	$\approx$	$1{,}10100110100110011001101 \times 2^7$
$7 + 127$	$=$	10000110

On obtient : 10000110 10100110100110011001101

Virgule flottante (suite)

- Écriture de 211,3

Décimal		Binaire
211,3	$\approx$	11010011,0100110011001101
	$\approx$	$1{,}10100110100110011001101 \times 2^7$
$7 + 127$	$=$	10000110

On obtient : 0 10000110 10100110100110011001101

Virgule flottante (suite)

- Écriture de 211,3

Décimal		Binaire
211,3	$\approx$	11010011,0100110011001101
	$\approx$	$1{,}10100110100110011001101 \times 2^7$
$7+127$	$=$	10000110

On obtient : 0 10000110 10100110100110011001101

- Valeur approchée : 211,300030517578125

Virgule flottante (suite)

- Écriture de 211,3

Décimal		Binaire
211,3	$\approx$	11010011,0100110011001101
	$\approx$	$1,10100110100110011001101 \times 2^7$
$7+127$	$=$	10000110

On obtient : 0 10000110 10100110100110011001101

- Valeur approchée : 211,300003051757812 5
- Précision de 7 chiffres minimum

Virgule flottante (suite)

- Écriture de 211,3

Décimal		Binaire
$211{,}3$	$\approx$	$11010011{,}01001100110011001101$
	$\approx$	$1{,}\textcolor{blue}{1010011010011001100110011 01}\times 2^7$
$7+127$	$=$	$\textcolor{red}{10000110}$

On obtient : $\textcolor{green}{0}$ $\textcolor{red}{10000110}$ $\textcolor{blue}{10100110100110011001101}$

- Valeur approchée : 211,300003051757812 5
- Précision de 7 chiffres minimum
- Imprécisions :
  - $3\times 0{,}1 \neq 0{,}3$
  - $0{,}1^2 \neq 0{,}01$
  - $(a+b)+c \neq a+(b+c)$

The magic trick

1	3	5	7	9	11	13	15
17	19	21	23	25	27	29	31
33	35	37	39	41	43	45	47
49	51	53	55	57	59	61	63

2	3	6	7	10	11	14	15
18	19	22	23	26	27	30	31
34	35	38	39	42	43	46	47
50	51	54	55	58	59	62	63

4	5	6	7	12	13	14	15
20	21	22	23	28	29	30	31
36	37	38	39	44	45	46	47
52	53	54	55	60	61	62	63

8	9	10	11	12	13	14	15
24	25	26	27	28	29	30	31
40	41	42	43	44	45	46	47
56	57	58	59	60	61	62	63

16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

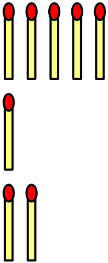
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Le jeu de Marienbad

- Variante du jeu de NIM apperçue dans “L'année dernière à Marienbad”
- Des allumettes sont placées en plusieurs tas ou piles.
- Deux personnes vont jouer à tour de rôle. Chacune peut enlever un nombre quelconque d'allumettes (au moins une) dans un et un seul tas de son choix.
- Celui qui prend la dernière allumette a gagné.

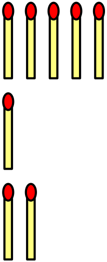
Stratégie gagnante

- Il faut mettre la "Nim-Addition" des nombres d'allumettes par tas à 0.



Stratégie gagnante

- Il faut mettre la “Nim-Addition” des nombres d'allumettes par tas à 0.



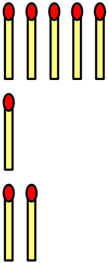
101

001

010

Stratégie gagnante

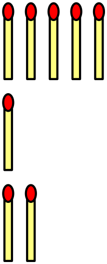
- Il faut mettre la “Nim-Addition” des nombres d'allumettes par tas à 0.



$$
\begin{array}{c}
101 \\
\oplus \\
001 \\
\oplus \\
010 \\
\hline
110
\end{array}
$$

Stratégie gagnante

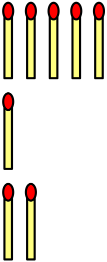
- Il faut mettre la "Nim-Addition" des nombres d'allumettes par tas à 0.



$$\begin{array}{ccc}
101 & \oplus & 110 \\
\oplus & & \\
001 & \oplus & 110 \\
\oplus & & \\
010 & \oplus & 110 \\
\hline
110 & &
\end{array}$$

Stratégie gagnante

- Il faut mettre la “Nim-Addition” des nombres d'allumettes par tas à 0.

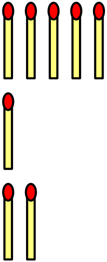


$$
\begin{array}{ccccc}
101 & \oplus & 110 & = & 011 \\
\oplus & & & & \\
001 & \oplus & 110 & = & 111 \\
\oplus & & & & \\
010 & \oplus & 110 & = & 100 \\
\hline
110 & & & &
\end{array}
$$

Stratégie gagnante

- Il faut mettre la "Nim-Addition" des nombres d'allumettes par tas à 0.

Il y a

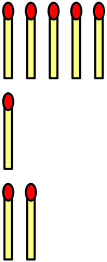


$$\begin{array}{ccccc} 101 & \oplus & 110 & = & 011 \\ \oplus & & & & \\ 001 & \oplus & 110 & = & 111 \\ \oplus & & & & \\ 010 & \oplus & 110 & = & 100 \\ \cline{1-1} 110 & & & & \end{array}$$

Stratégie gagnante

- Il faut mettre la “Nim-Addition” des nombres d'allumettes par tas à 0.

Il y a Il faut laisser

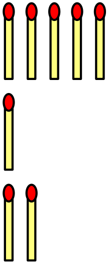


$$\begin{array}{ccccc} 101 & \oplus & 110 & = & 011 \\ \oplus & & & & \\ 001 & \oplus & 110 & = & 111 \\ \oplus & & & & \\ 010 & \oplus & 110 & = & 100 \\ \hline 110 & & & & \end{array}$$

Stratégie gagnante

- Il faut mettre la “Nim-Addition” des nombres d'allumettes par tas à 0.

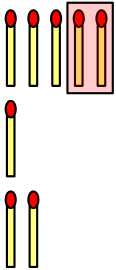
Il y a   Il faut laisser



$$\begin{array}{ccccc}
\boxed{101} & \oplus & 110 & = & \boxed{011} \\
\oplus & & & & \\
001 & \oplus & 110 & = & 111 \\
\oplus & & & & \\
010 & \oplus & 110 & = & 100 \\
\hline
110 & & & &
\end{array}$$

Stratégie gagnante

- Il faut mettre la “Nim-Addition” des nombres d'allumettes par tas à 0.



Il y a   Il faut laisser

$$
\begin{array}{ccccc}
\boxed{101} & \oplus & 110 & = & \boxed{011} \\
\oplus & & & & \\
001 & \oplus & 110 & = & 111 \\
\oplus & & & & \\
010 & \oplus & 110 & = & 100 \\
\hline
110 & & & &
\end{array}
$$

Stratégie gagnante

- Il faut mettre la “Nim-Addition” des nombres d'allumettes par tas à 0.