

Chapitre 2

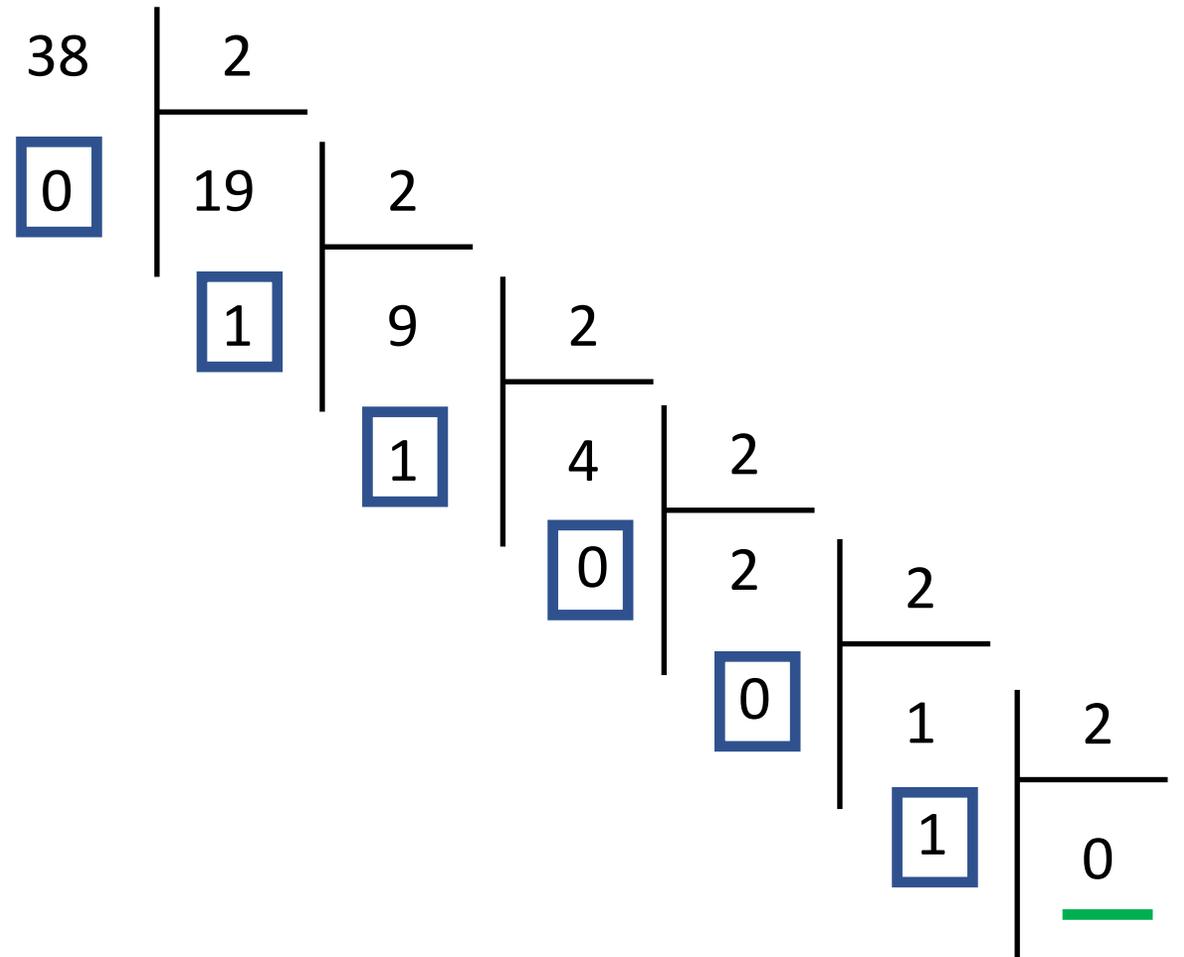
Représentation binaire d'un entier relatif

1^{ère} NSI

Thème 1 - Représentation des données : types et valeurs de base

Codage des entiers naturels (rappels)

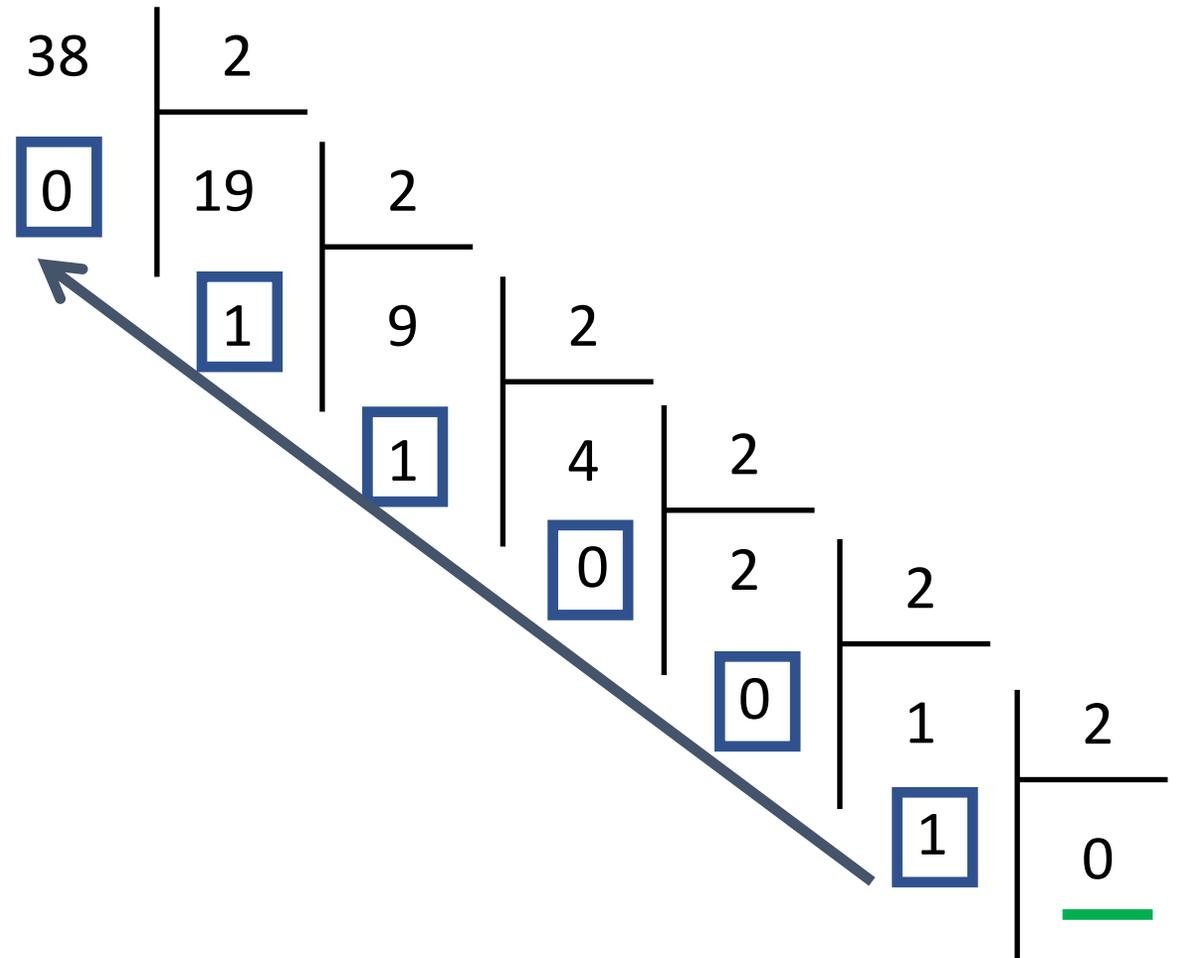
- Divisions successives par 2
- Exemple : 38 en binaire



Codage des entiers naturels (rappels)

- Divisions successives par 2
- Exemple : 38 en binaire

$$38 = 100110$$



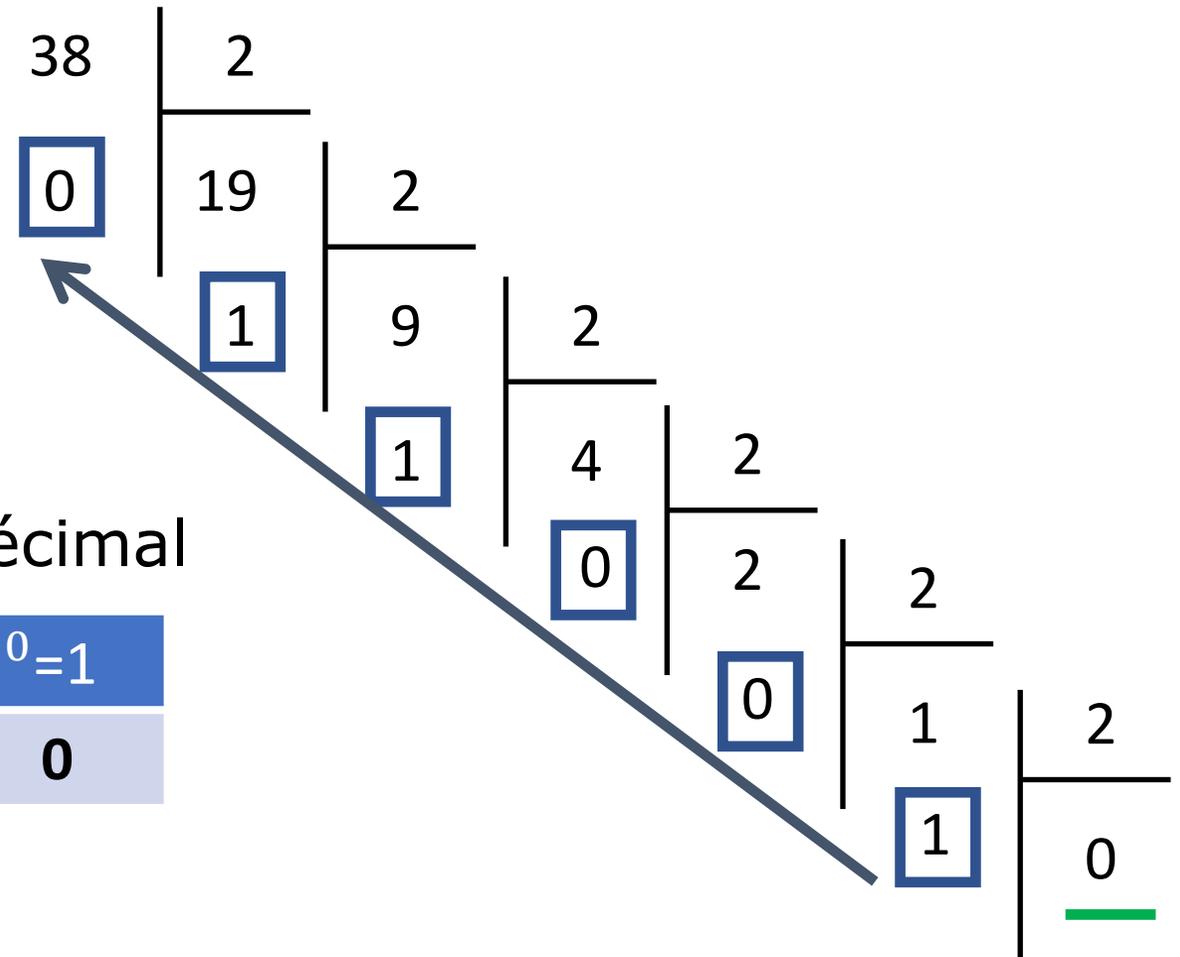
Codage des entiers naturels (rappels)

- Divisions successives par 2
- Exemple : 38 en binaire

$$38 = 100110$$

- Dans l'autre sens : 100110 en décimal

$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
1	0	0	1	1	0



Codage des entiers naturels (rappels)

- Addition binaire de deux entiers positifs :

- $(1)_2 + (1)_2 = (10)_2$

- Calcul de $(101)_2 + (1101)_2$:

$$\begin{array}{r} 101 \\ + 1101 \\ \hline 10010 \end{array}$$

$1 \ 1 \quad \leftarrow$ retenues

$10010 \quad \leftarrow$ résultat

Et pour les entiers relatifs ?

- On veut coder les entiers négatifs → comment ?

Et pour les entiers relatifs ?

- On veut coder les entiers négatifs → comment ?
- Sur 3 bits, on peut coder 8 valeurs ($2^3 = 8$).

000

001

010

011

100

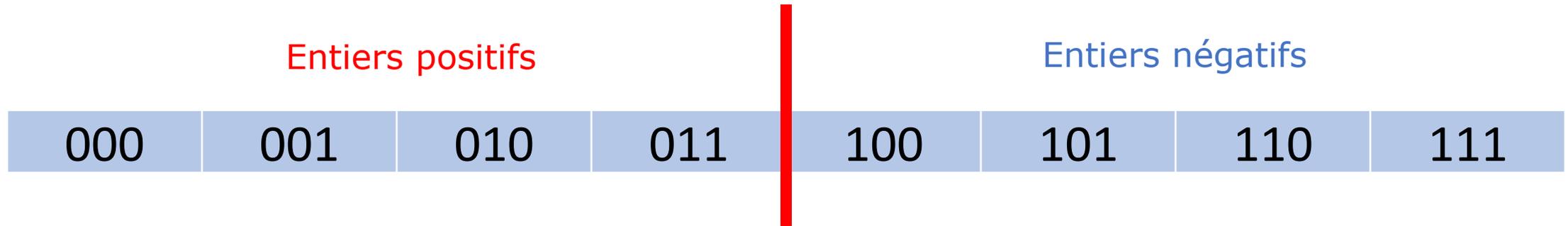
101

110

111

Et pour les entiers relatifs ?

- On veut coder les entiers négatifs → comment ?
- Sur 3 bits, on peut coder 8 valeurs ($2^3 = 8$).



- Idée : 4 valeurs → entiers positifs
4 valeurs → entiers (strictement) négatifs
- A quel entier (relatif) correspond chaque valeur binaire ?

Et pour les entiers relatifs ?

- **1^{ère} idée** : utiliser le premier bit pour le signe et le reste pour la valeur absolue de l'entier

Entiers positifs				Entiers négatifs			
000	001	010	011	100	101	110	111
+0	+1	+2	+3				

- Et la suite ?

Et pour les entiers relatifs ?

- **1^{ère} idée** : utiliser le premier bit pour le signe et le reste pour la valeur absolue de l'entier

Entiers positifs				Entiers négatifs			
000	001	010	011	100	101	110	111
+0	+1	+2	+3	-0	-1	-2	-3

Et pour les entiers relatifs ?

- **1^{ère} idée** : utiliser le premier bit pour le signe et le reste pour la valeur absolue de l'entier

Entiers positifs				Entiers négatifs			
000	001	010	011	100	101	110	111
+0	+1	+2	+3	-0	-1	-2	-3

- 1^{er} problème (mineur) : il existe deux zéros (+0 et -0)

Et pour les entiers relatifs ?

- **1^{ère} idée** : utiliser le premier bit pour le signe et le reste pour la valeur absolue de l'entier

Entiers positifs				Entiers négatifs			
000	001	010	011	100	101	110	111
+0	+1	+2	+3	-0	-1	-2	-3

- 1^{er} problème (mineur) : il existe deux zéros (+0 et -0)
- 2^{ème} problème (majeur) : l'algorithme d'addition ne marche plus

Et pour les entiers relatifs ?

- **1^{ère} idée** : utiliser le premier bit pour le signe et le reste pour la valeur absolue de l'entier

Entiers positifs				Entiers négatifs			
000	001	010	011	100	101	110	111
+0	+1	+2	+3	-0	-1	-2	-3

- 1^{er} problème (mineur) : il existe deux zéros (+0 et -0)
- 2^{ème} problème (majeur) : l'algorithme d'addition ne marche plus

Posons l'addition $2 + (-1)$:

$$\begin{array}{r} 010 \rightarrow 2 \\ + 101 \rightarrow -1 \\ \hline \mathbf{111} \rightarrow \mathbf{-3} \end{array}$$

Et pour les entiers relatifs ?

- **1^{ère} idée** : utiliser le premier bit pour le signe et le reste pour la valeur absolue de l'entier

Entiers positifs				Entiers négatifs			
000	001	010	011	100	101	110	111
+0	+1	+2	+3	-0	-1	-2	-3

- 1^{er} problème (mineur) : il existe deux zéros (+0 et -0)
- 2^{ème} problème (majeur) : l'algorithme d'addition ne marche plus

Posons l'addition $2 + (-1)$:

$$\begin{array}{r} 010 \rightarrow 2 \\ + 101 \rightarrow -1 \\ \hline \mathbf{111} \rightarrow \mathbf{-3} \end{array}$$

→ **MAUVAISE METHODE**

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2

Entiers positifs				Entiers négatifs			
000	001	010	011	100	101	110	111
0	1	2	3	?	?	?	?

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2

Entiers positifs				Entiers négatifs			
000	001	010	011	100	101	110	111
0	1	2	3	?	?	?	?

- Comment coder -3 ?
 - 1^{ère} étape : on passe en positif : 3

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2

Entiers positifs				Entiers négatifs			
000	001	010	011	100	101	110	111
0	1	2	3	?	?	?	?

- Comment coder -3 ?
 - 1^{ère} étape : on passe en positif : 3
 - 2^{ème} étape : on code 3 en binaire : 011

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2

Entiers positifs				Entiers négatifs			
000	001	010	011	100	101	110	111
0	1	2	3	?	?	?	?

- Comment coder -3 ?
 - 1^{ère} étape : on passe en positif : 3
 - 2^{ème} étape : on code 3 en binaire : 011
 - 3^{ème} étape : on inverse les bits : 100 (les 1 deviennent 0 et inversement)

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2

Entiers positifs				Entiers négatifs			
000	001	010	011	100	101	110	111
0	1	2	3	?	?	?	?

- Comment coder -3 ?

- 1^{ère} étape : on passe en positif : 3
- 2^{ème} étape : on code 3 en binaire : 011
- 3^{ème} étape : on inverse les bits : 100 (les 1 deviennent 0 et inversement)
- 4^{ème} étape : on ajoute 1 : 101

$$\begin{array}{r} 100 \\ + \quad 1 \\ \hline 101 \end{array}$$

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2

Entiers positifs				Entiers négatifs			
000	001	010	011	100	101	110	111
0	1	2	3	?	-3	?	?

- Comment coder -3 ?

- 1^{ère} étape : on passe en positif : 3
- 2^{ème} étape : on code 3 en binaire : 011
- 3^{ème} étape : on inverse les bits : 100 (les 1 deviennent 0 et inversement)
- 4^{ème} étape : on ajoute 1 : 101
- Conclusion : -3 se code 101

$$\begin{array}{r} 100 \\ + \quad 1 \\ \hline 101 \end{array}$$

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2

Entiers positifs				Entiers négatifs			
000	001	010	011	100	101	110	111
0	1	2	3	-4	-3	-2	-1

- Comment coder -3 ?
 - 1^{ère} étape : on passe en positif : 3
 - 2^{ème} étape : on code 3 en binaire : 011
 - 3^{ème} étape : on inverse les bits : 100 (les 1 deviennent 0 et inversement)
 - 4^{ème} étape : on ajoute 1 : 101
 - Conclusion : -3 se code 101
- On procède de même pour les autres valeurs

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2

Entiers positifs				Entiers négatifs			
000	001	010	011	100	101	110	111
0	1	2	3	-4	-3	-2	-1

- Il n'y a plus les problèmes de la 1^{ère} idée !
 - On n'a qu'un seul zéro !
 - L'algorithme d'addition fonctionne toujours :

$$\begin{array}{r} 010 \rightarrow 2 \\ + 101 \rightarrow -3 \\ \hline \mathbf{111} \rightarrow \mathbf{-1} \end{array}$$

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2

Entiers positifs				Entiers négatifs			
000	001	010	011	100	101	110	111
0	1	2	3	-4	-3	-2	-1

- Il n'y a plus les problèmes de la 1^{ère} idée !
 - On n'a qu'un seul zéro !
 - L'algorithme d'addition fonctionne toujours :

$$\begin{array}{r}
 010 \rightarrow 2 \\
 + 101 \rightarrow -3 \\
 \hline
 \mathbf{111} \rightarrow -\mathbf{-1}
 \end{array}$$

$$\begin{array}{r}
 \mathbf{11} \leftarrow \text{retenues} \\
 010 \rightarrow 2 \\
 + 111 \rightarrow -1 \\
 \hline
 \mathbf{1001} \rightarrow
 \end{array}$$

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2

Entiers positifs				Entiers négatifs			
000	001	010	011	100	101	110	111
0	1	2	3	-4	-3	-2	-1

- Il n'y a plus les problèmes de la 1^{ère} idée !
 - On n'a qu'un seul zéro !
 - L'algorithme d'addition fonctionne toujours :

$$\begin{array}{r}
 010 \rightarrow 2 \\
 + 101 \rightarrow -3 \\
 \hline
 \mathbf{111} \rightarrow -\mathbf{-1}
 \end{array}$$

$$\begin{array}{r}
 \mathbf{11} \leftarrow \text{retenues} \\
 010 \rightarrow 2 \\
 + 111 \rightarrow -1 \\
 \hline
 (1) \mathbf{001} \rightarrow 1 \text{ sur 3 bits}
 \end{array}$$

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2 → **Méthode utilisée !**
- Il est indispensable de préciser sur combien de bits on code les entiers ! (8 bits, 16 bits, 32 bits, 64 bits)
- Le bit de poids fort indique toujours le signe de l'entier : 0 pour un entier positif et 1 pour un entier négatif

Entiers positifs				Entiers négatifs			
000	001	010	011	100	101	110	111
0	1	2	3	-4	-3	-2	-1

Et pour les entiers relatifs ?

- **A vous de jouer !**
- Quelle est la représentation de -6 en complément à deux sur 4 bits ?
- Quelle est la représentation de -25 en complément à deux sur 8 bits ?

Méthode :

- 1°) on passe en positif
- 2°) on convertit le résultat en binaire (sur le nombre de bits disponibles)
- 3°) on inverse tous les bits
- 4°) on ajoute 1 pour avoir la réponse (sans tenir compte de la dernière retenue)

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2
- Quels entiers peut-on coder ?
 - Sur 3 bits : 8 entiers
 - 4 entiers positifs : de 0 à 3
 - 4 entiers strictement négatifs : de -4 à -1
 - Sur 8 bits ?

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2
- Quels entiers peut-on coder ?
 - Sur 3 bits : 8 entiers
 - 4 entiers positifs : de 0 à 3
 - 4 entiers strictement négatifs : de -4 à -1
 - Sur 8 bits ? 256 entiers
 - 128 entiers positifs : de 0 à 127
 - 128 entiers négatifs : de -128 à -1

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2
- Quels entiers peut-on coder ?
 - Sur 3 bits : 8 entiers
 - 4 entiers positifs : de 0 à 3
 - 4 entiers strictement négatifs : de -4 à -1
 - Sur 8 bits ? 256 entiers
 - 128 entiers positifs : de 0 à 127
 - 128 entiers négatifs : de -128 à -1
- Plage d'entiers codés ?
 - Sur 8 bits : de -128 à 127 = de -2^{8-1} à $2^{8-1} - 1$

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2
- Quels entiers peut-on coder ?
 - Sur 3 bits : 8 entiers
 - 4 entiers positifs : de 0 à 3
 - 4 entiers strictement négatifs : de -4 à -1
 - Sur 8 bits ? 256 entiers
 - 128 entiers positifs : de 0 à 127
 - 128 entiers négatifs : de -128 à -1
- Plage d'entiers codés ?
 - Sur 8 bits : de -128 à 127 = de -2^{8-1} à $2^{8-1} - 1$
 - Sur n bits : de -2^{n-1} à $2^{n-1} - 1$

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2
- Au fait, comment ça fonctionne dans l'autre sens ?
Quel est l'entier représenté par 10001011 en complément à deux sur 8 bits ?

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2
- Au fait, comment ça fonctionne dans l'autre sens ?
Quel est l'entier représenté par 10001011 en complément à deux sur 8 bits ?
 - Le bit de poids fort est un « 1 » (10001011) → il s'agit d'un entier négatif
 - On inverse tous les bits : 01110100
 - On ajoute 1 : 01110101
 - On passe en décimal : $1 + 4 + 16 + 32 + 64 = 117$
 - Conclusion : -117 est représenté par 10001011 en complément à deux sur 8 bits.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	1	1	0	1	0	1

Et pour les entiers relatifs ?

- **2^{ème} idée** : le complément à 2
- Au fait, comment ça fonctionne dans l'autre sens ?
Quel est l'entier représenté par 10001011 en complément à deux sur 8 bits ?
 - Le bit de poids fort est un « 1 » (10001011) → il s'agit d'un entier négatif
 - On inverse tous les bits : 01110100
 - On ajoute 1 : 01110101
 - On passe en décimal : $1 + 4 + 16 + 32 + 64 = 117$
 - Conclusion : -117 est représenté par 10001011 en complément à deux sur 8 bits.
- Et si le bit de poids fort est un 0 ?
→ on peut tout de suite calculer sa valeur décimale

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	1	1	0	1	0	1

Et pour les entiers relatifs ?

- **A vous de jouer !**
- Quel est l'entier représenté par 11101100 en complément à deux sur 8 bits ?

Méthode :

- Si le bit de poids fort est un 0, c'est un entier positif donc on sait faire (cf. chapitre 1)
- Si le bit de poids fort est un 1, c'est un entier négatif et on procède ainsi :
 - 1°) on inverse les bits
 - 2°) on ajoute 1
 - 3°) on convertit en décimal
 - 4°) on rajoute le signe « - » pour avoir la réponse

Conclusion

- Pour coder les entiers relatifs on utilise la représentation dite « en complément à deux »
- Il est nécessaire de savoir combien de bits sont utilisés
- Sur 8 bits (par ex.) on peut coder 256 entiers : la moitié seront des entiers positifs (de 0 à 127) et l'autre moitié des entiers (strictement) négatifs (de -128 à -1)
- On a vu :
 - la méthode pour déterminer la représentation en complément à deux d'un entier négatif (voir l'explication à la fin du cours pour comprendre pourquoi on fait les étapes évoquées)
 - la méthode pour faire déterminer l'entier relatif correspondant à une représentation en complément à deux