BROUILLON - QUELS DÉCALAGES ENTRE LES ENTIERS SIGNÉS ET LES NON SIGNÉS ?

CHRISTOPHE BAL

Document, avec son source LaTeX, disponible sur la page
`https://github.com/bc-writing/drafts`.

Mentions « légales »

Table des matières

1. Entiers non signés

Dans ce document, nous allons travailler uniquement avec des entiers non signés, c'est à dire des naturels, dont l'écriture en base 2 tient sur 8 bits même si les raisonnements tenus se généralisent à 16, 32 ou 64 bits. Ce choix tient juste à des contraintes de lisibilité du document.

Voici des exemples de représentations de type little-endian : on part à gauche de la puissance la plus haute, pour nous ce sera toujours 2^7, puis on va vers la plus basse *(c'est le même principe que l'écriture des naturels en base* 10*)*.

1	**0**	**0**	**0**	**0**	**1**	**0**	**1**	$\leftarrow 133 = 128 + 4 + 1 = 2^7 + 2^2 + 2^0$

0	**0**	**1**	**0**	**0**	**1**	**1**	**0**	$\leftarrow 38 = 32 + 4 + 2 = 2^5 + 2^2 + 2^1$

Avec 8 bits on peut représenter les naturels de 0 à $1 + 2 + 2^2 + \cdots + 2^7 = 2^8 - 1 = 255$.

Date: 17 Mars 2019.

2. Entiers signés

2.1. **Comment ça marche ?** Voici des exemples de représentations toujours sur 8 bits et de type little-endian où la case rouge indiquera toujours un bit de signe.

$$\begin{array}{|c|c|c|c|c|c|c|c|}\hline \mathbf{0}&\mathbf{1}&\mathbf{1}&\mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\\hline\end{array} \leftarrow 120 = 64+32+16+8 = \boxed{0\times\left(-2^7\right)} + 2^6+2^5+2^4+2^3$$

$$\begin{array}{|c|c|c|c|c|c|c|c|}\hline \mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\\hline\end{array} \leftarrow -120 = -128+8 = \boxed{1\times\left(-2^7\right)} + 2^3$$

$$\begin{array}{|c|c|c|c|c|c|c|c|}\hline \mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\\hline\end{array} \leftarrow -128 = \boxed{1\times\left(-2^7\right)} + 0$$

Avec 8 bits on peut représenter les naturels de $-2^7 = -128$ à $1+2+2^2+\cdots+2^6 = 2^7-1 = 127$.

Ce choix permet d'effectuer des additions relatives toujours sous forme d'addition bit à bit, et donc par conséquent de faire aussi des soustractions via des additions bit à bit. Voici quelques exemples sans dépassement de capacité.

- Addition de deux positifs.

$$\begin{array}{c|c|c|c|c|c|c|c|c|l}\cline{2-9} & \mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{0} & \leftarrow 84 = 64+16+4\\\cline{2-9} + & \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{1} & \leftarrow 17 = 16+1\\\cline{2-9} = & \mathbf{0}&\mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{1} & \leftarrow 101 = 64+32+4+1\\\cline{2-9}\end{array}$$

- Addition de deux négatifs *(la retenue finale hors capacité est ignorée)*.

$$\begin{array}{c|c|c|c|c|c|c|c|c|l}\cline{2-9} & \mathbf{1}&\mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{0} & \leftarrow -84 = \boxed{-128}+44 = \boxed{-128}+32+8+4\\\cline{2-9} + & \mathbf{1}&\mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{1}&\mathbf{1}&\mathbf{1}&\mathbf{1} & \leftarrow -17 = \boxed{-128}+111 = \boxed{-128}+64+32+8+4+2+1\\\cline{2-9} = & \mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{1}&\mathbf{1} & \leftarrow -101 = \boxed{-128}+27 = \boxed{-128}+16+8+2+1\\\cline{2-9}\end{array}$$

- Addition de deux entiers de signes différents.

$$\begin{array}{c|c|c|c|c|c|c|c|c|l}\cline{2-9} & \mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{0} & \leftarrow 84 = 64+16+4\\\cline{2-9} + & \mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{1}&\mathbf{1} & \leftarrow -101 = \boxed{-128}+27 = 16+8+2+1\\\cline{2-9} = & \mathbf{1}&\mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{1}&\mathbf{1}&\mathbf{1}&\mathbf{1} & \leftarrow -17 = \boxed{-128}+111 = -128+64+32+8+4+2+1\\\cline{2-9}\end{array}$$

- Autre addition de deux entiers de signes différents *(la retenue finale hors capacité est ignorée)*.

$$\begin{array}{c|c|c|c|c|c|c|c|c|l}\cline{2-9} & \mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{0} & \leftarrow 84 = 64+16+4\\\cline{2-9} + & \mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{1} & \leftarrow -61 = \boxed{-128}+67 = 64+2+1\\\cline{2-9} = & \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{1}&\mathbf{1}&\mathbf{1} & \leftarrow 23 = 16+4+2+1\\\cline{2-9}\end{array}$$

2.2. **Qu'est-ce qui motive ces choix ?** Pour comprendre comment découvrir ce type de procédé, nous allons raisonner en base 10 et imaginer que nous voulions calculer $189-32 = 157$ en utilisant uniquement des additions. Traditionnellement on fait comme suit.

$$\begin{array}{cccc} & 1 & 8 & 9\\ - & 0 & 3 & 2\\\hline = & 1 & 5 & 7\end{array}$$

Maintenant ajoutons 1000 à $189-32$. Nous obtenons :

$$\begin{array}{ccccc} & 0 & 1 & 8 & 9\\ - & 0 & 0 & 3 & 2\\ + & 1 & 0 & 0 & 0\\\hline = & 1 & 1 & 5 & 7\end{array}$$

Comme le résultat initial tenait sur les trois chiffres de droite, l'ajout de 1000 donne un nouveau résultat dont nous savons que les trois chiffres de droite sont ceux de $189 - 32$. D'autre part, $1000 + 189 - 32 = 189 + 968$ est une simple addition.

Notons aussi que $1000 - 32 = 999 - 032 + 1 = 967 + 1$ où 9 , 6 et 7 sont les compléments à 9 de 0 , 3 et 2 respectivement. L'opération $1000 - 32$ se fait donc rapidement. De plus, la soustraction $1157 - 1000$ revient à ignorer le 1 tout à droite. En résumé, nous avons fait comme suit où $\bullet$ indique une ligne où a été fait le calcul d'un complément à 9 plus 1 :

$$
\begin{array}{ccccc}
 & 0 & 1 & 8 & 9 \\
- & 0 & 0 & 3 & 2 \\
\hline\hline
 & 0 & 1 & 8 & 9 \\
+ & \bullet & 9 & 6 & 8 \\
\hline
= & 1 & 1 & 5 & 7 \\
\hline\hline
+ & & 1 & 5 & 7
\end{array}
$$

Très bien mais que se passe-t-il avec $32 - 189 = -157$ qui est négatif ? Testons pour voir.

$$
\begin{array}{ccccc}
 & 0 & 0 & 3 & 2 \\
- & 0 & 1 & 8 & 9 \\
\hline\hline
 & 0 & 0 & 3 & 2 \\
+ & \bullet & 8 & 1 & 1 \\
\hline
= & 0 & 8 & 4 & 3
\end{array}
$$

Oups ! Nous ne voyons pas directement 157. C'est normal puisque $843 - 1000 = -157$ n'a pas été effectué. Dans un tel cas, on note que le chiffre à gauche étant 0, on effectue un complément à 9 plus 1 tout en ajoutant un signe moins afin d'obtenir le résultat final. En résumé, nous procédons comme suit.

$$
\begin{array}{ccccc}
 & 0 & 0 & 3 & 2 \\
- & 0 & 1 & 8 & 9 \\
\hline\hline
 & 0 & 0 & 3 & 2 \\
+ & \bullet & 8 & 1 & 1 \\
\hline
= & 0 & 8 & 4 & 3 \\
\hline\hline
- & \bullet & 1 & 5 & 7
\end{array}
$$

Ne nous emballons pas trop vite car il reste un cas problématique que nous n'avons pas abordé. Examinons ce qu'il se passe avec $-32 - 189 = -221$?

$$
\begin{array}{ccccc}
- & 0 & 0 & 3 & 2 \\
- & 0 & 1 & 8 & 9 \\
\hline\hline
 & \bullet & 9 & 6 & 8 \\
+ & \bullet & 8 & 1 & 1 \\
\hline
= & 1 & 7 & 7 & 9
\end{array}
$$

Nous devons affiner notre règle car sinon ici nous aurions un résultat positif ! En fait, il est facile de voir qu'il suffit de comparer le nombre de compléments à 9 plus 1 effectués, c'est à dire le nombre de 1000 ajoutés, au chiffre tout à gauche qui sera 0 ou 1. Ici nous avons deux compléments et un chiffre 1 ce qui nous demande d'appliquer $2 - 1 = 1$ complément au résultat final tout en mettant un signe moins. En résumé, nous avons :

$$
\begin{array}{ccccc}
- & 0 & 0 & 3 & 2 \\
- & 0 & 1 & 8 & 9 \\
\hline\hline
 & \bullet & 9 & 6 & 8 \\
+ & \bullet & 8 & 1 & 1 \\
\hline
= & 1 & 7 & 7 & 9 \\
\hline\hline
- & \bullet & 2 & 2 & 1
\end{array}
$$

Nous trouvons bien $-32 - 189 = -221$. Que c'est beau ! Les deux cas suivants achèvent notre exploration expérimentale.

$$
\begin{array}{ccccc}
 & 0 & 0 & 3 & 2 \\
+ & 0 & 1 & 8 & 9 \\
\hline
= & 0 & 2 & 2 & 1 \\
\hline\hline
+ & & 2 & 2 & 1
\end{array}
\qquad\qquad
\begin{array}{ccccc}
- & 0 & 0 & 3 & 2 \\
+ & 0 & 1 & 8 & 9 \\
\hline\hline
 & \bullet & 9 & 6 & 8 \\
+ & 0 & 1 & 8 & 9 \\
\hline
= & 1 & 1 & 5 & 7 \\
\hline\hline
+ & & 1 & 5 & 7
\end{array}
$$

Exercice 1. *Dans la méthode ci-dessous, il existe des cas problématiques. Lesquels ?*

Exercice 2. *En laissant les cas problématiques de côté, démontrer le caractère général de la méthode que nous avons juste exposée via quelques exemples.*

2.3. **Pourquoi ça marche ?** Revenons à nos entiers signés écrits sur 8 bits. Reprenons le calcul de $84 - 101 = -17$ en nous inspirant de ce qui a été vu dans la section précédente avec la base 10. Ici $\bullet$ indique une ligne où a été fait le calcul d'un complément à 1 plus 1.

$$
\begin{array}{c|c|c|c|c|c|c|c|c|}
 & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\
- & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\
\hline\hline
 & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\
+ & \bullet & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\
\hline
= & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\
\hline\hline
- & \bullet & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}
\end{array}
$$

Tout s'éclaire ! Reprenons le cas de $84 - 61 = 23$ où l'on avait une retenue à ignorer.

$$
\begin{array}{c|c|c|c|c|c|c|c|c|}
 & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\
- & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\
\hline\hline
 & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\
+ & \bullet & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\
\hline
= & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\
\hline\hline
+ & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1}
\end{array}
$$

Avec le nouvel éclairage, nous notons qu'en fait les bits rouges ne servent qu'à indiquer s'il faut ou non effectuer un complément à 1 plus 1 en changeant de signe. Avec la façon de stocker les entiers signés, nous avons alors les correspondances suivantes où les entiers additionnés a et b sont dans $[-128\,;127]$ et leur somme aussi, ce qui revient à ne considérer que les cas de non dépassement de capacité. Ci-après, $\boxed{1}$ et les cases sur fond gris indiquent l'utilisation d'un complément à 1 plus 1 que nous avions noté $\boxed{\bullet}$ ci-dessus, et le symbole $\sim$ est pour le calcul de la somme sans le bit de signe qui lui entre en jeu dans la ligne avec le signe $=$.

$$
\begin{array}{cc}
 & \boxed{0}\boxed{\cdots} \\
- & \boxed{0}\boxed{\cdots} \\
\hline\hline
 & \boxed{0}\boxed{\cdots} \\
+ & \boxed{1}\boxed{\cdots} \\
\hline
\sim & \boxed{0}\boxed{\cdots} \\
\hline\hline
= & \boxed{1}\boxed{\cdots}
\end{array}
\qquad
\begin{array}{cc}
 & \boxed{0}\boxed{\cdots} \\
- & \boxed{0}\boxed{\cdots} \\
\hline\hline
 & \boxed{0}\boxed{\cdots} \\
+ & \boxed{1}\boxed{\cdots} \\
\hline
\sim & \boxed{1}\boxed{\cdots} \\
\hline\hline
= & \boxed{0}\boxed{\cdots}
\end{array}
\qquad
\begin{array}{cc}
- & \boxed{0}\boxed{\cdots} \\
- & \boxed{0}\boxed{\cdots} \\
\hline\hline
 & \boxed{1}\boxed{\cdots} \\
+ & \boxed{1}\boxed{\cdots} \\
\hline
\sim & \boxed{0}\boxed{\cdots} \\
\hline\hline
= & \boxed{0}\boxed{\cdots}
\end{array}
\qquad
\begin{array}{cc}
- & \boxed{0}\boxed{\cdots} \\
- & \boxed{0}\boxed{\cdots} \\
\hline\hline
 & \boxed{1}\boxed{\cdots} \\
+ & \boxed{1}\boxed{\cdots} \\
\hline
\sim & \boxed{1}\boxed{\cdots} \\
\hline\hline
= & \boxed{1}\boxed{\cdots}
\end{array}
$$

Soustraction 1 de deux naturels — *Soustraction 2 de deux naturels* — *Addition 1 de deux relatifs négatifs* — *Addition 2 de deux relatifs négatifs*

Les soustractions et l'addition 2 ne posent aucun souci. Par contre l'addition 1 est problématique avec son résultat positif. En fait cette addition contredit notre hypothèse de non dépassement de capacité. Nous allons voir pourquoi.

Le cas de l'addition 1 correspond à $(a\,;b) \in [-128\,;-1]^2$ tel que $(128+a)+(128+b) \in [0\,;127]$ soit $0 \leq 256+a+b \leq 127$ *i.e.* $-256 \leq a+b \leq -129$ ce qui correspond à un dépassement de capacité comme annoncé.

Ceci achève de démontrer la validité des procédés d'addition et de soustraction d'entiers signés dans les cas de non dépassement de capacité. Notez que le cas évident d'une addition de deux naturels a été omis, et aussi que $-a+b = b-a = b+(-a)$ et le fait qu'un changement de signe n'est autre qu'un complément à 1 plus 1 permettent de compléter les cas non indiqués ci-dessus.

Exercice 3. *Étudiez plus généralement le cas d'une base $b \in \mathbb{N}_{\geq 3}$ quelconque.*

3. Division par 2 et décalage à droite

3.1. **Entiers non signés.** Lorsque l'on calcule (n div 2) le quotient entier par défaut d'un naturel non signé n divisé par 2, il suffit de retirer son bit de poids faible. Voici un exemple.

1	1	0	1	0	1	1	1	$\leftarrow n_0 = 128+64+16+4+2+1 = 215$
0	1	1	0	1	0	1	1	$\leftarrow n_1 = n_0$ div $2 = 107$
0	0	1	1	0	1	0	1	$\leftarrow n_2 = n_1$ div $2 = 53$
0	0	0	1	1	0	1	0	$\leftarrow n_3 = n_2$ div $2 = 26$

Chaque étape correspond à un décalage vers la droite tout en complétant par un zéro à gauche.

3.2. **Entiers signés.** Examinons ce qu'il se passe lorsque l'on calcule (n div 2) le quotient entier par défaut d'un naturel signé $n<0$ divisé par 2.

1	1	0	1	0	1	1	1	$\leftarrow n_0 = -128+64+16+4+2+1 = -41$
1	1	1	0	1	0	1	1	$\leftarrow n_1 = n_0$ div $2 = -21 = -128+64+32+8+2+1$

On en déduit que quelque soit le signe de l'entier signé, le quotient entier par défaut correspond à un décalage vers la droite tout en conservant le bit de signe tout à gauche. Notez bien que l'on décale tous les bits y compris celui du signe. Démontrons nos affirmations.

Le cas $n \in [0\,;127]$ étant évident, considérons $n \in [-128\,;-1]$. La représentation de n s'obtient alors via $n = -2^7 + \sum\limits_{0 \leq k \leq 6} b_k\, 2^k$ avec chaque b_k dans $\{0\,;1\}$. Nous avons alors :

$$n \text{ div } 2 = -2^6 + \sum_{1 \leq k \leq 6} b_k\, 2^{k-1}$$

$$n \text{ div } 2 = -2 \times 2^6 + 2^6 + \sum_{0 \leq k \leq 5} b_{k-1}\, 2^k$$

$$n \text{ div } 2 = -2^7 + 2^6 + \sum_{0 \leq k \leq 5} b_{k-1}\, 2^k$$

La dernière identité correspond bien au décalage avec conservation de signe qui a été annoncé plus haut.

3.3. **Synthèse.** Nous constatons que nous avons deux types de décalage à droite. Dans le cas des entiers non signés, le décalage vers la droite est noté `>>>` et il est appelé *« décalage logique à droite »*. Dans le cas des entiers signés, le décalage vers la droite est noté `>>` et il est appelé *« décalage arithmétique à droite »*.

4. Multiplication par 2 et décalage à gauche

4.1. **Entiers non signés.** Lorsque l'on calcule $2n$ le double d'un naturel non signé n, il suffit de retirer son bit de poids fort tout en ajoutant un zéro comme nouveau bit de poids faible. Voici un exemple sans dépassement de capacité.

0	**0**	**0**	**1**	**0**	**1**	**1**	**1**	$\leftarrow n_0 = 16 + 4 + 2 + 1 = 23$
0	**0**	**1**	**0**	**1**	**1**	**1**	**0**	$\leftarrow n_1 = 2n_0 = 46$
0	**1**	**0**	**1**	**1**	**1**	**0**	**0**	$\leftarrow n_2 = 2n_1 = 92$
1	**0**	**1**	**1**	**1**	**0**	**0**	**0**	$\leftarrow n_3 = 2n_2 = 184$

Chaque étape correspond à un décalage vers la gauche tout en complétant par un zéro à droite.

4.2. **Entiers signés.** Examinons ce qu'il se passe lorsque l'on calcule $2n$ si $n < 0$ est un naturel signé. Voici un exemple sans dépassement de capacité.

1	**1**	**0**	**1**	**0**	**1**	**1**	**1**	$\leftarrow n_0 = -128 + 64 + 16 + 4 + 2 + 1 = -41$
1	**0**	**1**	**0**	**1**	**1**	**1**	**0**	$\leftarrow n_1 = 2n_0 = -82 = -128 + 32 + 8 + 4 + 2$

On effectue un décalage à gauche de tous les bits, y compris celui du signe, comme avec les entiers non signés. Mais est-ce toujours vrai ? Que se passe-t-il si le 2$^{\text{e}}$ bit à gauche est nul ? En fait, sous l'hypothèse de non dépassement de capacité, tout fonctionne sans souci comme nous allons l'expliquer proprement tout de suite.

Le cas $n \in [0\,;127]$ étant évident, considérons $n \in [-128\,;-1]$. La représentation de n s'obtient alors via $n = -2^7 + \sum\limits_{0 \leq k \leq 6} b_k\, 2^k$ avec chaque b_k dans $\{0\,;1\}$. Nous avons alors :

$$2n = -2^8 + \sum_{0 \leq k \leq 6} b_k\, 2^{k+1}$$

$$2n = -2^8 + b_6\, 2^7 + \sum_{1 \leq k \leq 6} b_{k-1}\, 2^k$$

$2n = (b_6 - 2)\, 2^7 + \sum_{1 \leq k \leq 6} b_{k-1}\, 2^k$

Si $b_6 = 1$, nous avons bien le décalage annoncé. Supposons donc que $b_6 = 0$. Dans ce cas, nous obtenons :

$S = \sum_{0 \leq k \leq 6} b_k\, 2^k$

$S = \sum_{0 \leq k \leq 5} b_k\, 2^k$

$S \leq \sum_{0 \leq k \leq 5} 2^k$

$S \leq 2^6 - 1$

Nous en déduisons que $n = -2^7 + S \leq -2^7 + 2^6 - 1 = -2^6 - 1$, puis ensuite $2n \leq -2^7 - 2 < -2^7$ ce qui est bien un cas de dépassement de capacité.

4.3. **Synthèse.** Dans le cas des entiers signés ou non, nous utilisons le même décalage vers la gauche qui est noté `<<` . Pour préciser sur quel type d'entiers on effectue un décalage, on parlera de *« décalage logique à gauche »* pour les entiers non signés, et de *« décalage arithmétique à gauche »* pour les entiers signés.