TD1 : Représentation des entiers en machine

« Il y a 10 types de personnes : celles qui savent écrire en binaire, et les autres »

Correction :
De façon générale : il faut rédiger et justifier les réponses. Il faut mettre en valeur (souligner et encadrer) les réponses.

Exercice 1 Pour les questions suivantes, Je ne vous demande pas forcément une valeur, vous pouvez donner la réponse sous forme de formule mathématique (accompagnée d'une justification).

1. Combien de valeurs différentes peut-on représenter sur 4 bits ?
2. Combien de valeurs différentes peut-on représenter sur n bits ?
3. Combien de fichiers différents peut-on écrire sur 15437 octets ?
4. On veut créer un type qui permet de représenter les lettres minuscules de 'a' à 'z'. Quel est le nombre minimal de bits nécessaires pour cela ?
5. Et si on veut aussi pouvoir représenter les lettres majuscules de 'A' à 'Z' ?
6. Et avec les chiffres de '0' à '9' ?

Exercice 2 Montrer le théorème suivant :

Théorème 1. *Soit $b \geq 2$ un entier. Pour tout entier $n \in \mathbb{N}$, il existe un entier $p \geq 0$ et des entiers $(n_k)_{0\leq k\leq p}$ dans $\{0,1,\dots,b-1\}$ tels que n peut s'écrire sous la forme*

$$n = \sum_{k=0}^{p} n_k b^k .$$

De plus, si l'on impose $n_p \neq 0$, cette écriture est unique.

(Dans le cas d'une écriture unique, on écrit $n = \overline{n_k \dots n_0}^b$ et on parle de *décomposition de n en base b*.)

Exercice 3 Donner la représentation en complément à 2 sur 8 bits des entiers suivants : 5, 9, -18, -48, -128, 129, 2023, -2023.

Correction :
On cherche des représentations sur 8 bits, on respecte donc les règles suivantes :

- seuls les entiers de l'intervalle $[-2^7, 2^7-1]$ sont représentables, donc 129, 2023 et -2023 ne sont pas représentables ;
- les entiers positifs sont représentés en base 2, avec le bit de poids fort à gauche, donc :

$5 = 2^2+2^0$: | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | $9 = 2^3+2^0$: | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

- les entiers négatifs sont représentés par le complément à 2 sur 8 bits de la représentation en base 2 de leur valeur absolue ; on commence donc par donner la représentation en base 2 de la valeur absolue, puis on applique l'algorithme pour trouver le complément à 2 sur 8 bits, donc :
  - pour -18 : On a

$$18 = 2^4+2^1 : \boxed{0}\boxed{0}\boxed{0}\boxed{1}\boxed{0}\boxed{0}\boxed{1}\boxed{0}.$$

On applique le complémentaire bit à bit :

$$\boxed{1}\boxed{1}\boxed{1}\boxed{0}\boxed{1}\boxed{1}\boxed{0}\boxed{1},$$

puis on ajoute 1, en propageant la retenue mais en restant sur 8 bits :

$$\boxed{1}\boxed{1}\boxed{1}\boxed{0}\boxed{1}\boxed{1}\boxed{1}\boxed{0} : -18 .$$

– de même :

$48 = 2^5 + 2^4$: | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |

↓ compl. bit à bit

| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |

↓ +1sur 8 bits

| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | : −48

– et :

$128 = 2^7$: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

↓ compl. bit à bit

| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

↓ +1sur 8 bits

| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | : −128

(On remarque ici que le codage initial de 128 est un codage en base 2 mais n'est pas le codage en base 2 sur 8 bits, car le bit de signe est à 1.)

Exercice 4 Quels entiers (en base 10) sont représentés en complément à 2 sur 8 bits de la façon suivante :

- 00111001
- 11000010
- 01100110
- 10001101

Correction :
Les encodages commençant par 0 encode des entiers positifs, il suffit de les lire en base 2, donc :

$$00111001 : 2^5 + 2^4 + 2^3 + 2^0 = 57 \qquad 01100110 : 2^6 + 2^5 + 2^2 + 2^1 = 102\,.$$

Ceux commençant par 1 encode un nombre négatif. Il faut appliquer l'algorithme de l'exercice précédent en appliquant les étapes dans l'ordre inverse, donc :

$$11000010 \xrightarrow{-1} 11000001 \xrightarrow{\text{compl. bit à bit}} 00111110\,,$$

donc la valeur absolue de l'entier représenté est $2^5 + 2^4 + 2^3 + 2^2 + 2^1 = 62$ et l'entier représenté est -62.
De même :

$$10001101 \xrightarrow{-1} 10001100 \xrightarrow{\text{compl. bit à bit}} 01110011\,,$$

donc la valeur absolue de l'entier représenté est $2^6 + 2^5 + 2^4 + 2^1 + 2^0 = 115$ et l'entier représenté est -115.

Exercice 5

1. Montrer que l'algorithme donnée en cours pour trouver la représentation en complément à 2 d'un entier (complément bit à bit suivi d'une incrémentation) est effectivement correct.

Correction :
J'ai vu essentiellement deux versions de cette preuve dans les copies. Les deux fausses et utilisant du vocabulaire non introduit en classe. Quitte à me répéter : aller copier des corrections ne sert à rien, ça me fait perdre mon temps, ça me mets de mauvaise humeur, et vous n'avez rien appris.
Quand vous arrivez à une étape où vous écrivez que la somme de deux entiers strictement positifs vaut 0, vous êtes en droit de vous poser des questions et d'exercer votre esprit critique, plutôt que de continuer comme si de rien parce que ça vous permet de conclure.
Avant de faire une preuve, il faut comprendre ce qu'il y a à montrer. Ici, il faut montrer que l'algorithme donné en cours (et rappelé dans l'exercice) permet d'aboutir au complément à 2 sur N bits d'un nombre, *tel que défini en cours*. On a défini en cours le complément à 2 sur N bits de l'entier $k \in [-2^N, -1]$ comme l'entier obtenu en gardant les N bits de poids faible de $2^N - k$. Il faut donc montrer que l'algorithme aboutit à cet entier. Notons $\tilde{k}$ cet entier.
Soit un entier N fixé et un entier $k \in [2^{N-1}, -1]$. Notons k' l'entier obtenu en prenant le complément bit à bit de k sur N bits, et k'' l'entier sur $N+1$ bits obtenu en ajoutant 1 à k'. On a clairement que $k + k'$ est un

entier sur N bits dont tous les bits sont à 1 :

$$k + k' = \sum_{i=0}^{N-1} 2^i = 2^N - 1 ,$$

donc

$$k + k'' = k + k' + 1 = 2^N .$$

Ce qui permet d'obtenir

$$k'' = 2^N - k .$$

On voit que $\tilde{k}$ est l'entier obtenu en gardant les N bits de poids faible de k'', ce qui permet de conclure.

2. Quelle relation numérique y a-t-il entre deux entiers dont les représentations en compléments à 2 sur 8 bits ne diffèrent que par le bit de signe ?

Exercice 6 Soit un entier $b \geq 2$. Montrer que la décomposition en base b d'un entier $n > 0$ contient $1 + \lfloor log_b(n) \rfloor$ chiffres. *Indice : commencer par le montrer pour un entier de la forme* b^k.