RND. Nombres entiers. Exercices. Corrigés

***Exercice 1* : Vrai / Faux**
Q.1 : VRAI. Multiplier un nombre en binaire par 2 revient en effet à ajouter un 0 à sa droite.
Q.2 : FAUX. On utilise les nombres de 0 à 7.
Q.3 : VRAI. On peut représenter les entiers positifs de 0 à 2^n-1.
Q.4 : VRAI. Le symbole A vaut 10 en écriture hexadécimale. AA = $10 \times 16^0 + 10 \times 16^1 = 170$.
Q.5 : FAUX. On peut représenter les nombres de -2^3 à $2^3 - 1$ inclus soient de -8 à -7.
Q.6 : FAUX. L'entier 10 est bien codé par 01010 sur 5 bits. On inverse les bits 01010 => 10101 puis on ajoute 1, d'où le résultat 10110

***Exercice 2* : QCM**
Q.1 : Réponse 2 (au début des années 40).
Q.2 : Réponse 4. $1000 = 3 \times 16^2 + 14 \times 16^1 + 8 \times 16^0$ soit 3E8 en écriture hexadécimale soit 11 1110 1000 en binaire (on peut aussi remarquer que 2^3 divise 1000 d'où les 3 zéros en écriture binaire).
Q.3 : Réponse 3. Tout nombre pair est divisible par 2 donc se termine par 0 en écriture binaire.
Q.4 : Réponse 1. 1001 x 111 = 1000 x 111 + 111 = 111111.
Q.5 : Réponse 4. 2 est codé 00010 sur 5 bits. On inverse les bits 00010 => 11101 puis on ajoute 1, d'où le résultat 11110. Une autre approche était de partir de -1 qui est représenté par 11111 et d'enlever 1.

***Exercice 3* : Effectuer** *en binaire les additions des nombres entiers positifs (écrits en binaire) suivants :*

a) 111011
b) 11110101
c) 110001010
d) 111110 (on peut remarquer qu'il s'agit de 2 x 11111 et donc ajouter un 0 à droite de 11111).

***Exercice 4* : Déterminer** *les nombres négatifs suivants :*
Les 3 nombre binaires présentés ayant un bit fort de 1 sont bien des nombres négatifs.
Il faut d'abord retrancher 1 puis inverser chaque bit et enfin convertir le nombre obtenu en base 10.

a) 10001101 => 10001100 => 01110011 avec 01110011 = $1 \times 2^0 + 1 \times 2^1 + 1 \times 2^4 + 1 \times 2^5 + 1 \times 2^6 = 115$. Le nombre représenté est donc -115.
b) 11100010 => 11100001 = > 00011110 avec 00011110 = $1 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 1 \times 2^4 = 30$. Le nombre représenté est donc -30.
c) 11111111 => 11111110 = > 00000001 soit 1. Le nombre représenté est -1. Cela est normal car tout binaire composé uniquement de 1 représente -1.

***Exercice 5* : Additionner à l'aide du complément à 2^n.**

a) 12 en écriture binaire se représente par 1100 et -53 par 11001011.
b) 1100 + 11001011 = 11010111
On s'attend à trouver 12 + (-53) soit -41. 11010111 étant un nombre négatif (bit fort de 1), on cherche le nombre qu'il représente en base 10. Il faut d'abord retrancher 1 puis inverser chaque bit et enfin convertir le nombre obtenu en base 10.
c) 11010111 => 11010110 => 00101001 soit 41. 11010111 représente bien -41.

***Exercice 6* : Coder sur 16 bits des nombres relatifs.**
On peut représenter les nombres de -2^{15} à $2^{15} - 1$ inclus.

***Exercice 7* : Déterminer la taille des nombres entiers positifs en bases 2 et 16.**

a) Il y a chaque fois 2 possibilité par bit (0 ou 1), l'ordre étant important.
Pour 2 bits, 2 x 2 possibilités (nombres de 0 à 3 inclus) , 3 bits 2 x 2 x 2 possibilités (nombres de 0 à 7 inclus) et pour 8 bits 2 x 2 x … x 2 =2^8 possibilités (de 0 à 255 inclus).

b) Le mieux est de tester les puissances de 2 successives dans un tableau :

2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}
2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768

Il faut donc 15 bits pour représenter 19999 possibilités.
Remarque : pour les élèves poursuivant les mathématiques en terminale (spécialité ou complémentaire), vous verrez le logarithme qui permet de résoudre le problème simplement.

***Exercice 8 (*)* : Des résultats d'additions étranges.**

***Expliquer** pourquoi sur 8 bits on obtient les résultats suivants :*

a) 127 s'écrit 01111111 en binaire. 0111 1111 + 1 = 1000 0000 qui est un nombre négatif puisque son bit fort est de 1. Il faut d'abord retrancher 1 puis inverser chaque bit et enfin convertir le nombre obtenu en base 10.
1000 0000 => 0111 1111 = > 1000 0000 soit … 128. Ainsi, 128 et -128 sont les mêmes nombres, ainsi 128 est son propre opposé !
Remarque : sur un cercle trigonométrique, un angle de 180° ou de -180° correspondent également au même point
b) 127 + 2 = 127 + 1 + 1 = -128 + 1 =-127 (question précédente) d'où le résultat.
c) 127 + 127 = 128 – 1 + 128 – 1 = -2 (car 128 + 128 = 0, question a))

***Exercice 9 (*)* : Expliquer le bug de l'an 2038.**

Sur 32 bits, on peut coder les nombres entiers relatifs entre -2^{31} et 2^{31} – 1. Cela permet donc de coder 2^{31} – 1 secondes en tout.
1 an = 365,25 jours
1 jour = 24 heures
1 heure = 3600 secondes.
On convertit les 2^{31} – 1 secondes en années soient (2^{31} – 1)/(3600 x 24 x 365,25) où environ 68 ans.
1970 + 68 = 2038 d'où le bug prévisible de l'an 2038.
A noter que le système d'exploitation a déjà prévu un patch en prévision de ce bug !
Plus d'informations : https://www.lemondeinformatique.fr/actualites/lire-la-version-56-du-noyau-linux-corrige-le-bug-de-l-an-2038-78137.html