REPRÉSENTATION DES NOMBRES RELATIFS - EXERCICES

Exercice 1

On peut représenter les entiers de -8 à 7 (bien noter que l'ordre n'est pas habituel : cela est du à la représentation des négatifs par complément à deux)

Codage binaire (méthode du complément à 2)				Entier relatif
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	-8
1	0	0	1	-7
1	0	1	0	-6
1	0	1	1	-5
1	1	0	0	-4
1	1	0	1	-3
1	1	1	0	-2
1	1	1	1	-1

Exercice 2

Codage de -1	Codage de -56
• On écrit 1 en binaire : 00000001 • On inverse les bits : 11111110 • On rajoute 1 : 11111111	• On écrit 56 en binaire : 00111000 • On inverse les bits : 11000111 • On rajoute 1 : 11001000

Exercice 3

a) 01101100 commence par un 0, donc il est positif : on le décompose avec la méthode vue au chapitre 1 : $(01101100)_{c2} = (01101100)_2 = (108)_{10}$

b) 11101101 commence par un 1, il sera donc négatif. On applique la méthode du complément à 2 « à l'envers » :

- On retranche 1 : $(11101100)_2$
- On inverse tous les bits : $(00010011)_2$
- On convertit en base 10 en n'oubliant pas le signe « - » : $(-19)_{10}$

c) 1010101010101010 commence par un 1, il sera donc négatif. On applique la méthode du complément à 2 « à l'envers » :

- On retranche 1 : $(1010101010101001)_2$
- On inverse tous les bits : $(0101010101010110)_2$
- On convertit en base 10 en n'oubliant pas le signe « - » : $(-21\ 846)_{10}$

Exercice 4

Sur 2 octets, on a 16 bits, donc l'étendue des nombres possibles est $[-2^{15} ; 2^{15}-1]$ c'est-à-dire : [-32768, 32767]

Exercice 5

$(00010100)_{c2} + (11110001)_{c2} = (00000101)_{c2}$ avec la dernière retenue qui disparaît par manque de place.

Exercice 6

$(11001001)_{c2} = (-55)_{10}$ et $(0110110)_{c2} = (54)_{10}$

Exercice 7 : un peu de programmation !!!

Voir fichier *Complementa2.py*