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)

Mot binaire				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

Méthode	Exemple -8
On écrit le nombre en valeur absolue en binaire (→ complément à 1)	$(8)_{10}=(00001000)_2$ sur 8 bits
On inverse les bits de cette écriture ($0 \leftrightarrow 1$)	$(11110111)_2=(247)_{10}$
On ajoute 1 donc $(00000001)_2$ sur 8 bits en tenant compte des éventuelles retenues (1+1=1 0)	$(11111000)_2=(248)_{10}$ et 248 – 256 = - 8

Exercice 3

a) $(4)_{10}=(100)_2$ C'est la partie entière.

Traiton la partie décimale $(0{,}125)_{10}$ 0,125 x 2 = 0,25 = **0** + 0,25

0,25 x 2 = 0,5 = **0** + 0,5

0,5 x 2 = 1 = **1** + 0 **la partie décimale est nulle : stop !**

$(4{,}125)_{10}=(100{,}001)_2$

b) $(100{,}0101)_2=1\times2^2+0\times2^1+0\times2^0+0\times2^{-1}+1\times2^{-2}+0\times2^{-3}+1\times2^{-4}=(4{,}3125)_{10}$

c) $(0{,}1)_{10}$ 0,1 x 2 = 0,2 = **0** + 0,2 *ligne 1*

0,2 x 2 = 0,4 = **0** + 0,4 *ligne 2*

0,4 x 2 = 0,8 = **0** + 0,8 *ligne 3*

0,8 x 2 = 1,6 = **1** + 0,6 *ligne 4*

0,6 x 2 = 1,2 = **1** + 0,2 *ligne 5*

0,2 x 2 = 0,4 = **0** + 0,4 On se retrouve comme à la ligne 2

0,4 x 2 = 0,8 = **0** + 0,8 Comme la ligne 3

0,8 x 2 = 1,6 = **1** + 0,6 Comme la ligne 4

0,6 x 2 = 1,2 = **1** + 0,2 Comme la ligne 5

0,2 x 2 = 0,4 = **0** + 0,4 On se retrouve comme à la ligne 2

0,4 x 2 = 0,8 = **0** + 0,8 Comme la ligne 3 ….......etc.

Le processus de "conversion" $(0{,}1)_{10}$ ne s'arrête pas, nous obtenons : "0,0001100110011...", le schéma "0011" se répète à "l'infini". $(0{,}1)_{10}$ ne peut être représenté EXACTEMENT !

d) et e) Voir programme *Decimal.py*

Exercice 4

a) **1 0100 0110 1001 0000 0000 0000 0000 000**

Le premier 1 représente s = 1, le signe sera $(-1)^1$ donc **-**

0100 0110 est le n de l'exposant : $n=2^6+2^2+2^1=70$ d'où e=n-127 = 70-127= -57

1001 0000 0000 0000 0000 000 est la mantisse m et vaut :

m=1 (**bit implicite à ne pas oublier**)+ $2^{-1}+2^{-4}=1{,}5625$

Le nombre codé en base 10 est $-1{,}5625.2-57\approx1{,}0842.10-17$: c'est approximatif.

b)

0 0111 1100 0100 0000 0000 0000 0000 000 = $1{,}25.2^{-3}$ = 0,15625

Cet autre exemple est exact cette fois ci !