Exercice1

```
>>> 0b100110
38
>>> 0b001101
13
>>> 0b100110 + 0b001101
51
```

Exercice2

```
>>> int("1142",5)
172
>>> int("AC",16)
172

>>> int("B3",16)
179
>>> int("10110011",2)
179
#179=172+7
>>> int("1204",5)
179
```

Exercice3

```
def lendian(octet1,octet2,chaine):
   if chaine=="BE":
      return (octet1*16**2+octet2)
   if chaine=="LE":
      return (octet2*16**2+octet1)

print(lendian(0x08,0x00,"LE"))
print(lendian(0x08,0x00,"BE"))
```

Exercice4

```
>>> 0b1001
9
>>> 2**5-10
22
>>> bin(22)
'0b10110'#codage de -10
```

Le codage sur 5 bits permet de coder les entiers relatifs de -16 (10000) à 15 (01111).

Exercice5

$3,625=2+1+0,5+0,125=(11,101)_2$

Exercice 6

1)
$-4,5=-1,125 \times 2^2$
le signe se code avec 1
l'exposant décalé vaut $1025=(10000000001)_2$
la mantisse tronquée vaut $(0010\ldots00)_2$
Au final : 1100000000010010……..00.

2)
1011111111101000 0000 ... 0000
Il s'agit d'un nombre négatif
>>> 0b01111111110
1022
donc la puissance est -1
La mantisse tronquée est 1000 0000 ... 0000 ce qui correspond à 0,5
Au final il s'agit du réel : $-1,5\times2^{-1}= - 0,75$

Exercice 7

a	b	a and not(b)	not(a) and b	(a and not(b)) or (not(a) and b)
1	1	0	0	0
1	0	1	0	1
0	1	0	1	1
0	0	0	0	0