REPRÉSENTATION DES NOMBRES RELATIFS - CORRECTION

Exercice 1

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 2

a) 01101100 commence par un 0, donc il est positif : on le décompose avec la méthode vue au chapitre 1 : $(01101100)_{c2} = (01101100)_2 = 64+32+8+4 = (108)_{10}$

b) 11101101 commence par un 1, il sera donc négatif. On applique la méthode du complément à 2 :
- On inverse tous les bits : 00010010
- On ajoute 1 : 00010011
- On convertit en base 10 en n'oubliant pas le signe « - » :
  $(00010011)_2 = 16+2+1 = (19)_{10}$, donc $(11101101)_{c2} = (-19)_{10}$

c) 1010101010101010 commence par 1 donc il est négatif. On applique la méthode du complément à 2 :
- On inverse tous les bits : 0101010101010101
- On ajoute 1 : 0101010101010110
- On convertit en base 10 en n'oubliant pas le signe « - » :
  $0101010101010110 = 2^{14}+2^{12}+2^{10}+2^8+2^6+2^4+2^2 = 16384+4096+1024+256+64+16+4 = 21\ 846$, donc $(1010101010101010)_{c2} = (-21\ 846)_{10}$

Exercice 3

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

Exercice 4

Avec la méthode du complément à 2, on trouve que $(20)_{10}=(00010100)_{c2}$ et que $(-15)_{10}=(11110001)_{c2}$

Donc : $(20-15)_{10} = (20)_{10}+(-15)_{10} = (00010100)_{c2} + (11110001)_{c2} = (00000101)_{c2}$ avec <u>la dernière retenue qui disparaît par manque de place</u>.

Exercice 5

- 11001001 commence par un 1, il sera donc négatif :
  - On inverse tous les bits : 00110110
  - On ajoute 1 : 00110111
  - On convertit en base 10 en n'oubliant pas le signe « - » :
    $00110111 = 32+16+4+2+1 = 55$, donc $(11001001)_{c2} = (-55)_{10}$

- 0110110 commence par un 0, donc il est positif : on le décompose avec la méthode vue au chapitre 1 : $(0110110)_{c2} = (0110110)_2 = 32+16+4+2 = (54)_{10}$

<u>Exercice 6 : pour les fans de programmation !!!</u>

```python
def conversion(n):
    result=''
    q=n
    while q!=0:
        result=str(q%2)+result
        q=q//2
    return result
```

```python
def cpt1(mot):
    motcpt1 = ""
    for i in mot:
        if i == '0':
            motcpt1 += '1'
        else:
            motcpt1 += '0'
    return motcpt1
```

```python
def comp32b(mot):
    while len(mot)!=32:
        mot='0'+mot
    return mot

#OU :
def comp32b(mot):
    for i in range(32-len(mot)):
        mot='0'+mot
    return mot
```

```python
def cpt2(mot):
    taille = len(mot)
    retenue = 1
    motcpt2 = ""
    for j in range(-1,-taille-1,-1):
        if retenue == 1:
            if mot[j] == '0':
                motcpt2 = '1' + motcpt2
                retenue = 0
            else:
                motcpt2 = '0' + motcpt2
        else:
            motcpt2 = mot[j] + motcpt2
    return motcpt2
```

```python
mot=int(input("Choisir un nombre entier relatif : "))
if mot==0:
    print(mot,"s'écrit sur 32 bits avec la méthode du complément à 2 : ", "00000000000000000000000000000000" )

elif mot>0:
    mot2=conversion(mot) #on convertit le nombre de la base 10 vers la base 2
    mot232=comp32b(mot2) # on complète si besoin le nombre binaire en 32 bits
    print(mot,"s'écrit sur 32 bits avec la méthode du complément à 2 : ", mot232 )

else:
    mot2=conversion(-mot) #on convertit le nombre de la base 10 vers la base 2
    mot232=comp32b(mot2) # on complète si besoin le nombre binaire en 32 bits
    motcp1=cpt1(mot232) #on fait le complément à 1
    motcp2=cpt2(motcp1) #on fait le complément à 2
    print(mot,"s'écrit sur 32 bits avec la méthode du complément à 2 : ", motcp2)
```