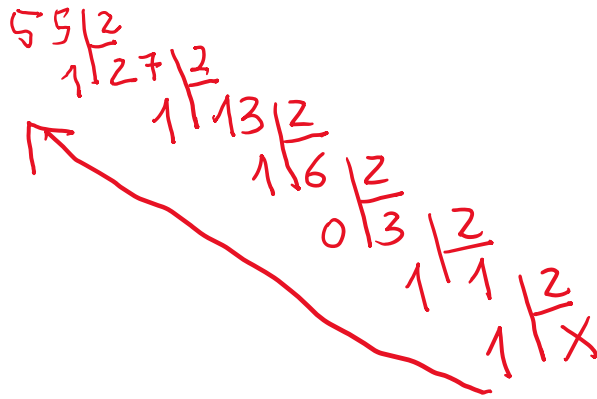


Q1 + 55 en C2 sur 8 bits



+55 est positif il suffit de le mettre au format de 8 bits : $(0011\ 0111)_{c2}$

Q2 -40 en C2 sur 8 bits

- 40 est négatif on part du nombre positif mis au format de 8 bits

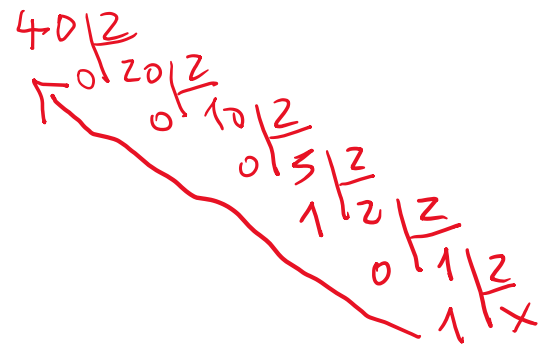
+40 : 0010 1000

Retenue 11

C1 : 1101 0111

C2 = C1+1 + 1

$$\begin{array}{r} 1101\ 1000 \\ \uparrow \quad \uparrow \quad \uparrow \\ -128 + 64 + 16 + 8 = -40 \end{array} \qquad (-40) : (1101\ 1000)_{c2}$$



Q3 55 + (-40) en C2 sur 8 bits

Retenue 1 111

+55 : 0011 0111

+

(-40) : 1101 1000

 0000 1111

↑ dépasse du format

Donc le résultat est 0000 1111 soit +15

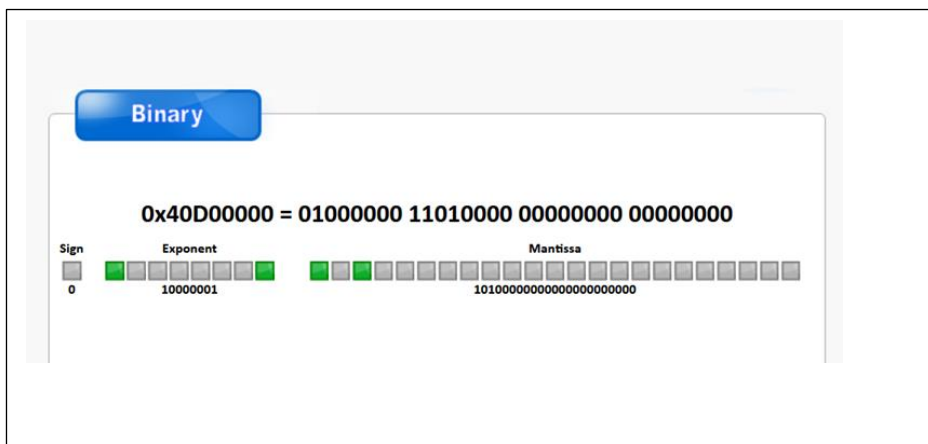
Q4) $10,375 \rightarrow 10 = (1010)$
 $0,375 \times 2 = 0,750$
 $0,75 \times 2 = 1,50$
 $0,5 \times 2 = 1,0$
 $(1010,011)_2$

$0,375$
 \downarrow
 $0,5 \quad 0,250 \quad 0,125$
 $\downarrow \quad \downarrow$
 $(1010,011)_2$

Q5) $\Lambda_{10} \times 2^{10}$

$2 + 127 = 129 \rightarrow 1000\ 0001$

$$\begin{array}{r} \text{S} \\ \hline 0 \quad 1000\ 0001 \quad 101\ 0000\ 0000\ 0000\ 0000\ 0000 \end{array}$$



Q6)
$$\begin{array}{l} A \quad (41)_{16} = (4 \times 16 + 1) = (65)_{10} \\ a \quad (61)_{16} = (6 \times 16 + 1) = (97)_{10} \end{array} \Bigg)_{32}$$

$$\begin{array}{l} \underline{z} \quad (5A)_{16} = 5 \times 16 + 10 = (90)_{10} \\ \gamma \quad (7A)_{16} = (7 \times 16) = (112)_{10} \end{array} \Bigg)_{32}$$

Q7) $m = M + 32$

for i in range(65, 90) :

$chr(i) \quad chr(i+32)$

Q8) $u + 03B1 \quad 0011 \quad 1011 \mid 0001$

1	1	0	D10	D9	D8	D7	D6	1	0	D5	D4	D3	D2	D1	D0
1	1	0	0	1	1	1	0	1	0	1	1	0	0	0	1

C
l
E
l
B
) 1

Codage	hex	dec (bytes)	dec	binary
UTF-8	CE B1	206 177	52913	11001110 10110001

```

1 def affiche():
2     for i in range(65,91):
3         print(f" Majuscule : {chr(i)} minuscule :
4         {chr(i+32)}")
5 affiche()
6
7 # def affiche():
8 #     for i in range(???,???):
9 #         print(f" Majuscule : {chr(???????)}
10 #         minuscule : {chr(???????????)}")
11 #
12 # affiche()
13
14

```

Python

This is the Pyzo interpreter with integrated event
Type 'help' for help, type '?' for a list of *mag:
Running script: "I:\Branly24\NSI\Première\Chapitre
Numération\DS_Numération.py"

Majuscule : A minuscule : a
Majuscule : B minuscule : b
Majuscule : C minuscule : c
Majuscule : D minuscule : d
Majuscule : E minuscule : e
Majuscule : F minuscule : f
Majuscule : G minuscule : g
Majuscule : H minuscule : h
Majuscule : I minuscule : i
Majuscule : J minuscule : j
Majuscule : K minuscule : k
Majuscule : L minuscule : l
Majuscule : M minuscule : m
Majuscule : N minuscule : n