

I - Codage d'une date sur seulement 2 octets

Une date est composée de 3 éléments :

- le jour est un nombre entre 1 et 31, codé sur 5 bits
- le mois est un nombre entre 1 et 12, codé sur 4 bits
- l'année sera codée pour optimiser le format. Afin d'économiser de l'espace mémoire, on prend l'initiative de définir un **nombre auquel il faudra ajouter 1980** pour retrouver l'année codée (ex : 12 correspond à 1992). L'année est codée sur 7 bits

- Donner la date codée dans la variable suivante : D= 0b1100110110001000 (jour, mois, année).
- Coder la date d'aujourd'hui avec ce principe.
- Donner les scripts Python des fonctions capables de décoder et de coder ces informations.

On donne ici les opérateurs sur les bits en Python

$x \ll y$ # Décalage vers la gauche y fois

$x \gg y$ # Décalage vers la droite y fois

$x \& y$ # ET bit à bit

$x | y$ # OU bit à bit

$x \wedge y$ # OU exclusif

II - Codage des grandeurs réelles : 32 bits à virgule flottante.

La donnée est un nombre signé (+/-) sur 32 bits (4 octets) à virgule flottante respectant la norme IEEE 754. Elle est formée de trois éléments ; la mantisse (nombre fractionnaire) sur 23 bits, l'exposant (décalé de la valeur 127) sur 8 bits et le bit de signe (S = 1 pour un nombre négatif) et sont répartis comme suit :

Rang	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	S	Exposant								Mantisse																						
Poids		2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}	2^{-16}	2^{-17}	2^{-18}	2^{-19}	2^{-20}	2^{-21}	2^{-22}	2^{-23}

Le nombre a pour valeur : $(-1)^S \times (1 + \text{mantisse}) \times 2^{(\text{exposant} - 127)}$

Exemple

Soit le nombre 00111110 00100000 00000000 00000000

En isolant les bits on trouve le format suivant : 0 01111100 010000000000000000000000,

Le signe est nul, l'exposant est $124 - 127 = -3$, et la mantisse est 0,25 ($0 \times 2^{-1} + 1 \times 2^{-2}$). Le nombre représenté est donc $+1,25 \times 2^{-3}$, ce qui donne + 0,15625.

- Exprimer en base 10 le nombre réel de codage IEEE 754 qui est codé en hexadécimal avec la valeur suivante : 0xC2D5A000
- Exprimer en base 10 les nombres IEEE 754 ci-après :
 - 0b01000000010010010000111111010000
 - 0b10111101010000000000000000000000

[illegible][illegible][illegible]

Convertisseur IEEE 754 (JavaScript), V0.22

	Pancarte	Exposant	Mantisse
Évaluer:	+1	1	1.5707950592041016
Encodé comme :	0	2 128	4788176
Binaire:	<input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Tu es entré	3.14159		
Valeur réellement stockée dans float :	3.141590118408203125		
Erreur due à la conversion :	1.18408203125E-7		
Représentation binaire	01000000010010010000111111010000		
Représentation hexadécimale	0x40490fd0		

+1
-1