1 Écriture d'un nombre décimal

- 1. Décomposition d'un nombre :
 - (a) En notation décimale, décomposer le nombre $254,\!36$ en faisant apparaı̂tre explicitement ses puissances de 10 (positives ou négatives).

 $254,36 = 2 \times 10^3 + \dots$

- (b) Faire la même chose en notation binaire, avec le nombre 1101,011 : $1101,011=\dots$
- 2. Convertir les nombres suivants en notation binaire :

a) 12,6875

- b) 13, 325
- c) 7,09375
- 3. Convertir les nombres suivants en notation décimale :

a) $(1001, 1011)_2$

- b) (10101, 011101)₂
- 4. Écrire en notation scientifique les nombres suivants :

a) $(-105, 745)_{10}$

- b) $(0.0745)_{10}$
- c) $(1011, 0111101)_2$
- d) $(0,0000001101)_2$
- 2 Représentation des nombres à virgule flottante en norme IEEE-754 simple précision (32 bits)

Rappel de la norme : un nombre s'écrit $N=s\times m\times 2^e$:

- le bit de poids fort (MSB) donne le signe $(s = \pm 1) : 0$ pour positif et 1 pour négatif.
- les 8 bits suivants donnent l'exposant e, décalé de +127.
- les 23 derniers bits donnent la partie fractionnaire de la mantisse $m \ (1 \le m < 10)$
- 1. Trouver la représentation de :

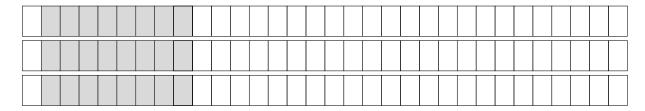
(a) 1011,0111101: (b) 0.0000001101:



- 2. Trouver les nombres décimaux représentés par :
 - (a) 1 10000101 1111000110000000000000000
- 3. Au format simple précision de la norme IEEE-754, écrire les nombres suivants :

a) 2^{-23}

- b) -4,0
- c) 0,25



4	0.1	+	0.2	= 0	1.3 '

(a)	Co	mp	léte	er l	a r	epr	ése	nta	tio	n d	u r	on	ıbre	e de	écir	nal	0,	1:							
(b)	En	dé	dui	<u>re</u> l	la 1	epi	rése	enta	atic	n (lu 1	non	nbr	e d	éci	ma	10,	,2:							

(c) Expliquer pourquoi python indique:

False

5. Donner la représentation du nombre rationnel 1/3.



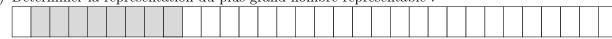
6. Deux cas limites:

(a) Déterminer la représentation du plus petit nombre positif représentable :



Donner une approximation de sa valeur en base décimale.

(b) Déterminer la représentation du plus grand nombre représentable :



Donner une approximation de sa valeur en base décimale.

7. TP : contrôler tous les résultats avec Python ou les convertisseurs en ligne :

https://www.h-schmidt.net/FloatConverter/IEEE754.html

http://www.binaryconvert.com/result_float.html

Remarques complémentaires :

— l'infini (positif ou négatif) se code avec un exposant 11111111 et une mantisse nulle :

— les codes où l'exposant vaut 11111111 et la mantisse est non nulle servent à coder des erreurs

NaN (Not a Number):

(exple de NaN): 0 11111111 00010011100101000000000