## I – Écriture d'un nombre décimal

- 1. Décomposition d'un nombre :
  - (a) En notation décimale, décomposer le nombre 704,36 en faisant apparaître explicitement ses puissances de 10 (positives ou négatives).

 $704.36 = 7 \times 10^2 + \dots$ 

- (b) Faire la même chose en notation binaire, avec le nombre 1101,011 :  $1101,011 = \dots$
- 2. Donner l'écriture binaire des nombres suivants :

a)  $(12,6875)_{10}$ 

b)  $(13,3125)_{10}$ 

- c)  $(7,09375)_{10}$
- 3. Donner l'écriture décimale des nombres suivants :

a) (1001, 1011)<sub>2</sub>

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

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

b)  $(0,0745)_{10}$ 

c) (1011, 0111101)<sub>2</sub>

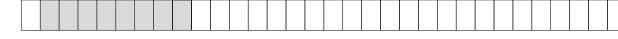
d)  $(0,0000001101)_2$ 

## II – 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)  $(1011, 01111101)_2 = (11, 4765625)_{10}$ :



(b)  $(0,0000001101)_2 = (0,0126953125)_{10}$ :

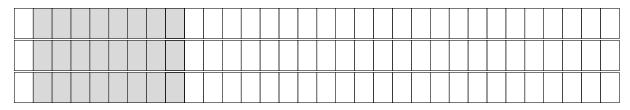


- 2. Trouver les nombres décimaux représentés par :
  - (a) 1 10000101 1111000110000000000000000
- 3. Au format simple précision de la norme IEEE-754, écrire les nombres suivants :

a)  $2^{-23}$ 

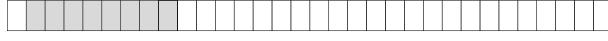
b) -4,0

c) 0,25



4	0.1	- 1	0.2 =	. n 2?
4	11.1	_		

(a	) Co	mp	léte	er l	a r	epr	ése	nta	tio	n d	u 1	on	ıbr	e d	éciı	mal	0,	1:
			_															



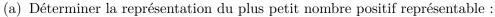
(c) Expliquer pourquoi python indique:

False



			1											/															
Г																													$\overline{}$
											ı					ı													1 1
																ı	l	l						l					í I
											ı	l .				ı													1 1
																ı	l	l						l					1 1
											ı					ı													1 1
								l .			ı					ı													1 1
														-							-				-				

6. Deux cas limites:





Donner une approximation de sa valeur en base 10.

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



Donner une approximation de sa valeur en base 10.

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 111111111 00010011100101000000000

(P1.III - TD)