Lycée Jean Moulin — **Draguignan**
NSI 1re — Année ....

Représentation des réels

Les décimaux

Définition :
Un décimal est un nombre à virgule dont la partie décimale est finie.
• On peut l'écrire : $d = \dfrac{N}{10^p}$ où $N \in \mathbb{Z}$ et $p \in \mathbb{N}$
Par exemple : $15{,}325=\dfrac{15325}{10^3}$
• On peut le décomposer comme dans l'exemple :
$15{,}325 = 1\times10^1 + 5\times10^0 + 3\times10^{-1} + 2\times10^{-2} + 5\times10^{-3}$

Les réels

Un nombre réel est un nombre à virgule (ou pas) et dont la partie décimale est finie ou pas.
• -2 est un réel
• 15,325 est un réel
• $\dfrac{2}{3}=0{,}666666...$ est un réel
• $\sqrt{2}$ est un réel

En informatique...

En informatique tous les nombres sont des décimaux, et ce car on est limité en mémoire...
On ne peut coder un nombre exactement si sa partie décimale est infinie.

Comment coder un nombre réel ?

Il faut pour cela le décomposer en puissance de deux:
Prenons un exemple : 15,3
La partie entière 15 s'écrit en binaire 1111
Pour la partie décimale, on procède comme suit :

$$
\begin{aligned}
0{,}3 \times 2 &= \boxed{0}{,}6\\
0{,}6 \times 2 &= \boxed{1}{,}2\\
0{,}2 \times 2 &= \boxed{0}{,}4\\
0{,}4 \times 2 &= \boxed{0}{,}8\\
0{,}8 \times 2 &= \boxed{1}{,}6
\end{aligned}
$$

On multiplie la partie décimale par 2
On reporte la partie décimale obtenue que l'on multiplie par 2 et ainsi de suite.
On s'arrête quand la précision est satisfaisante, dans le cas présent, on voit bien que l'on peut continuer indéfiniment...

On obtient :
$0{,}3 = \boxed{0}\times2^{-1} + \boxed{1}\times2^{-2} + \boxed{0}\times2^{-3} + \boxed{0}\times2^{-4} + \boxed{1}\times2^{-5} + \boxed{1}\times2^{-6} + ...$
L'écriture en binaire de 15,3 est : $1111{,}0\,\underbrace{\mathbf{1001}}\,\underbrace{1001}\,\underbrace{1001}\,...$

Remarquez la partie qui se répète indéfiniment...

Pour preuve :
$0{,}3 = (\boxed{0} + 0{,}6)\times 2^{-1} = (\boxed{0} + (\boxed{1} + 0{,}2)\times 2^{-1})\times 2^{-1} = \boxed{0}\times 2^{-1} + \boxed{1}\times 2^{-2} + 0{,}2\times 2^{-2}$
En remplaçant 0,2 par $(\boxed{0} + 0{,}4)\times 2^{-1}$, puis 0,4 par $(\boxed{0} + 0{,}8)\times 2^{-1}$
et enfin 0,8 par $(\boxed{1} + 0{,}6)\times 2^{-1}$
On obtient bien la décomposition écrite plus haut.

? Exercice 1 :

Donner les écritures binaires des nombres

- 17,2
- 35,31
- 5,25
- 0,5
- 128,375
- 0,05

Les nombres rationnels dyadiques

Définition :
Un nombre rationnel dyadique (ou fraction dyadique) est un rationnel qui peut s'écrire sous la forme : $\dfrac{N}{2^p}$ où $N \in \mathbb{Z}$ et $p \in \mathbb{N}$
De tels nombres ont des écritures binaires finies.

Par exemple : 0,5 est dyadique

$0{,}5 \times 2 = \boxed{1}{,}0$
$0{,}0 \times 2 = \boxed{0}{,}0$

L'écriture binaire de 0,5 est 0,1

? Exercice 2 :

Montrer que ces nombres sont dyadiques

- $\dfrac{3}{8}$
- 12,75
- 2,375

Du binaire au décimal

Par exemple pour convertir en décimal le nombre binaire 101,11
On a : $1\times 2^2 + 0\times 2^1 + 1\times 2^0 + 1\times 2^{-1} + 1\times 2^{-2} = 4 + 1 + 0.5 + 0.25 = 5{,}75$

? Exercice 3 :

Convertir en décimal le nombre binaire 101001,10011

Notation en virgule flottante

La notation en virgule flottante (ou notation scientifique) permet de normaliser l'écriture des nombres réels, elle a donné son nom au type **float** associé aux nombres à virgule en informatique.

Décimal		Binaire
211	=	11010011
2.11×10^2	=	1.1010011×2^7
0.05	=	0.0000110011
5×10^{-2}	=	1.10011×2^{-5}

MÉTHODE :

Un nombre réel se représente sur 32 bits par :

$$s\overbrace{\underbrace{eeeeeeee}_{8\text{ bits}}}^{\text{exposant}}\overbrace{\underbrace{mmmmmmmmmmmmmmmmmmmmmmm}_{23\text{ bits}}}^{\text{mantisse}}$$ Avec:

- 1 bit pour le signe (s). 1 pour (-) et 0 pour (+)
- 8 bits pour l'exposant (On ajoute 127 à la puissance de 2 pour éviter d'avoir à coder un nombre négatif)
- 23 bits pour la mantisse (le code binaire du nombre lui-même hormis le 1 avant la virgule qui est toujours présent)

On a donc : Nombre $= (-1)^s \times 1{,}\text{mantisse} \times 2^{\text{exposant}-127}$

EXEMPLE :

Écriture de 211.3

Décimal		Binaire
211.3	$\approx$	11010011.0100110011001101
	$\approx$	$1.10100110100110011001101 \times 2^7$
$7+127$	=	10000110

On obtient : 0 10000110 10100110100110011001101

Valeur approchée : 211.3000030517578125

EXERCICE 4 :

Donner l'écriture binaire sur 32 bits de 35,31

On ne compare pas l'égalité de deux flottants...

Du fait de l'imprécision du codage de nombre réels non dyadiques, le test d'égalité entre flottants ne donne pas toujours le résultat escompté. Pour contourner ce problème on compare leur différence à un nombre proche de zéro.

```
>>> 0.1+0.2==0.3
False
>>> 0.1+0.2-0.3<10**(-10)
True
```