RD04 REPRÉSENTATION DES FLOTTANTS

Pour mémoire, les nombres flottants sont ce que nous appelons en mathématiques les nombres réels ($\mathbb{R}$). A ceci près qu'en Python, on ne peut pas représenter tous les réels mais uniquement les nombres qui auront un nombre fini de chiffres après la virgule. Dans les autres cas, nous devrons faire des approximations.

I. Les nombres dyadiques

Définition 1 : les nombres décimaux

Un nombre décimal est un nombre s'écrvant sous la forme $\frac{x}{10^n}$ où x est un entier relatif et n un entier naturel. Visuellement, un nombre décimal possède un nombre fini de chiffres après la virgule.

Nombre	4	-3	0.25	1/3	1/4	π	10/3
Décimal?	OUI	OUI	OUI	NON	OUI	NON	NON

Définition 2 : les nombres dyadiques

Par analogie, on appelle nombres dyadiques les nombres s'écrivant sous la forme $\frac{x}{2^n}$.

Exercice 1 : Dire si les nombres suivants sont dyadiques

Nombre	13/4	1/5	2.5	25	10/3	π	3.125
Dyadique?							

Définition 3 : Développement dyadique d'un nombre

On appelle développement dyadique d'un nombre l'écriture binaire de ce nombre.

Propriété 1 : Écriture binaire d'un nombre dyadique

Pour obtenir le développement dyadique d'un nombre qui peut s'écrire sous la forme $\frac{x}{2^n}$, on prend le nombre binaire correspondant à x et on insère une virgule avant le n-ième bit en partant de la fin.

Exemple : Écrire le développement dyadique de 2.5

$2.5 = \frac{5}{2^1}$. Le nombre binaire correspondant à 5 est 101_2.
En insérant une virgule avant le premier bit en partant de la fin, on obtient 10.1_2.

Exercice 2 : Trouver les représentations binaires des nombres dyadiques suivants : $\frac{17}{4}$ et $\frac{54}{64}$

II. Conversion d'un nombre décimal en base 2

Méthode 1

1. Conversion de la partie entière du nombre décimal
2. Multiplication de la partie décimale par 2
3. Récupération de la partie entière
4. Si la partie décimale est nulle on arrête sinon retour au 2.
5. Assemblage de la partie entière et des bits récupérés pour former l'écriture binaire

Exemple : Conversion du nombre 5.1875 **en binaire.**

1. Conversion de la partie entière : $5_{10} = 101_2$
2. Multiplications successives par 2 de la partie décimale jusqu'à ce qu'elle soit nulle.

$$\begin{aligned} 0.1875 \times 2 = \quad & 0.375 \quad && = \underline{0} + 0.375 \\ 0.375 \times 2 = \quad & 0.75 \quad && = \underline{0} + 0.75 \\ 0.75 \times 2 = \quad & 1.5 \quad && = \underline{1} + 0.5 \\ 0.5 \times 2 = \quad & 1 \quad && = \underline{1} + 0.0 \end{aligned}$$

On assemble ensuite la partie entière et la partie décimale, ainsi : $5.1875 = 101{,}0011_2$

Exercice 3 : Convertir les nombres suivants :

9.6875 2.625

Remarque

Il existe des nombres décimaux non dyadiques ! Ainsi si on applique la méthode ci-dessus, alors le développement ne s'arrêtera pas. C'est pour cela qu'il faudra définir une précision souhaitée. Et cela explique aussi certains comportements étranges au premier abord de Python.

```
>>> print(0.1 + 0.1)
0.2
>>> print(0.1 + 0.1 + 0.1)
0.30000000000000004
```

- En Python, les flottants sont codés sur 64 bits
- Il faut éviter de tester l'égalité de deux flottants.

Exercice 4 : Expliquer par le calcul le résultat de l'addition $1.1 + 0.1$

III. De l'écriture binaire à la notation décimale

Exemple : Convertir $100{,}0101_2$

1. Convertir la partie entière : $100_2 = 4_{10}$
2. Convertir la partie décimale. On utilise un tableau des puissances de 2 négative.

Rang du bit	-1	-2	-3	-4	-5	-6
Puissance de 2	$2^{-1} = 0.5$	$2^{-2} = 0.25$	$2^{-3} = 0.125$	$2^{-4} = 0.0625$	$2^{-5} = 0.03125$	$2^{-6} = 0.015625$
Bit	0	1	0	1	0	0
Valeur	0	0.25	0	0.0625	0	0

Il suffit ensuite d'additionner toutes les valeurs calculées : $100{,}0101_2 = 4 + 0.25 + 0.0625 = 4.3125_{10}$

Exercice 5 : Trouver la représentation décimale de 101.001

1. Convertir la partie entière : .....
2. Convertir la partie décimale. On utilise un tableau des puissances de 2 négative.

Rang du bit	-1	-2	-3	-4	-5	-6
Puissance de 2	$2^{-1} = 0.5$	$2^{-2} = 0.25$	$2^{-3} = 0.125$	$2^{-4} = 0.0625$	$2^{-5} = 0.03125$	$2^{-6} = 0.015625$
Bit						
Valeur						

Exercice 6 : Convertir en décimal les nombres suivants :

1101,100101 1000,11101

IV. Notation scientifique

En base dix, il est possible d'écrire les très grands nombres et les très petits nombres grâce aux "puissances de dix" (exemples $6{,}02 \times 10^{23}$ ou $6{,}67 \times 10^{-11}$).

Il est possible de faire exactement la même chose avec une représentation binaire, puisque nous sommes en base 2, nous utiliserons des "puissances de deux" à la place des "puissances dix" (exemple $101{,}1101.2^{10}$).

Exemple : Pour passer d'une écriture sans "puissance de deux" à une écriture avec "puissance de deux", il suffit décaler la virgule : $1101{,}1001 = 1{,}1011001.2^{11}$.

Pour passer de 1101,1001 à 1,1011001 nous avons décalé la virgule de 3 rangs vers la gauche d'où le 2^{11}. (attention de ne pas oublier que nous travaillons en base 2 le " 11 " correspond bien à un décalage de 3 rangs de la virgule, car $11_2 = 3_{10}$.

Exemple : Si l'on désire décaler la virgule vers la gauche, il va être nécessaire d'utiliser des "puissances de deux négatives". $0{,}0110_2 = 1{,}10.2^{-10}$. Nous décalons la virgule de 2 rangs vers la droite, d'où le " -10 ".

Exercice 7 : Trouver l'écriture binaire scientifique de :

1. $1011{,}0111101_2$ .....
2. 0.0745_{10} .....