Informatique PCSI
TP 12a : nombres flottants

Exercice 1

L'expression `0.2 + 0.1 > 0.3` a la valeur `True`. Quelle en est la raison ?

1. C'est une erreur de la machine.
2. C'est parce-que la machine n'utilise que 32 bits au lieu de 64 pour coder les flottants.
3. C'est parce-que 0,1 ne peut pas être représenté en flottant de manière exacte.
4. C'est parce-que > signifie $\geq$ pour l'interpréteur Python.

Exercice 2

Le programme qui suit calcule 2^i puis 1.01^i pour 30000 valeurs de i entier ; deviner, avant de tester le programme sur machine, quel sera le calcul le plus rapide ?

```
from time import time

top = time()
for i in range(30000):
    c = 2 ** i
print(time() - top)

top = time()
for i in range(30000):
    c = 1.01 ** i
print(time() - top)
```

Tester le programme. Quelle peut être l'explication du comportement observé ?

Exercice 3

On représente des nombres réels avec une écriture signe, mantisse, exposant en binaire. On utilise six bits $b_0b_1b_2b_3b_4b_5$. Le bit b_0 représente le signe : 0 pour $+$, 1 pour $-$. Les bits $b_1b_2b_3$ représentent la mantisse tronquée. Les bits b_4b_5 représentent l'exposant. Par exemple, l'écriture 010010 représente un nombre positif. Sa mantisse vaut 1,100 soit $1,5$ en décimal et l'exposant 10 vaut 2. Le nombre représenté est donc $1,5 \times 2^2$ soit 6.

1. Quel est le plus petit nombre positif et le plus grand nombre que l'on peut écrire ainsi ?
2. Combien de nombres appartenant l'intervalle $[1; 2[$ peut-on écrire ?
3. Quel est le successeur de 2 ?

Exercice 4

Les flottants sont utilisés pour des calculs approchés. L'algorithme de Briggs par exemple est employé pour le calcul d'une valeur approchée du logarithme d'un nombre.

Il est basé sur l'approximation : $\log u \simeq u - 1$ pour u suffisamment proche de 1. Si $x^{1/2^n} - 1 \simeq 0$, alors $\log x^{1/2^n} \simeq x^{1/2^n} - 1$, d'où $(1/2^n) \log x \simeq x^{1/2^n} - 1$. On en déduit : $\log x \simeq 2^n(x^{1/2^n} - 1)$.

Programmer cet algorithme et le tester avec quelques valeurs.

Exercice 5

Pour calculer une valeur approchée de $\ln 2$, l'algorithme de Brouncker utilise l'égalité $\ln 2 = \int_1^2 \ 1/x \ \mathrm{d}x$. Un découpage sous la courbe entre $x = 1$ et $x = 2$ permet après calculs d'obtenir une valeur approchée de l'intégrale : $\sum_{k=1}^{n} \frac{1}{(2k-1)2k}$. Programmer cet algorithme.