

Informatique PCSI

TP 12a corrigé : nombres flottants

Exercice 1

La somme des deux flottants $0.2 + 0.1$ donne 0.30000000000000004 . C'est la réponse 3.

Exercice 2

Il semble plus simple de calculer 2^i que 1.01^i . Mais Python peut calculer avec des entiers très grands et pour les calculs de 2^i , une grande partie du temps est utilisée pour stocker et manipuler ces entiers. Pour le calcul de 1.01^i , Python utilise des valeurs approchées et calcule donc très rapidement. (Plus d'une seconde pour le calcul des 2^i et 0,01 s pour le calcul des 1.01^i).

Exercice 3

1. Le plus petit nombre positif s'écrit 0000000, c'est 1 ($1,0 \times 2^0$).
Le plus grand nombre s'écrit 011111, c'est 15 ($1,875 \times 2^3$).
2. La mantisse est dans l'intervalle $[1; 2[$. Donc le signe doit être positif ($b_0 = 0$) et l'exposant doit être nul ($b_4b_5 = 00$). La mantisse tronquée $b_2b_3b_4$ peut prendre huit valeurs de 000 à 111. Il y a donc huit nombres appartenant à $[1; 2[$. Le plus petit est 1, le plus grand est 1,875.
3. Le nombre 2 s'écrit $1,0 \times 2^1$, donc est représenté par 000010. Son successeur est représenté par 000110 qui représente $1,125 \times 2^1 = 2,25$.

Exercice 4

```
def briggs(x):
    n = 0
    while abs(x - 1) > 0.0001:
        x = x ** 0.5
        n = n + 1
    x = x - 1
    return x * 2 ** n

from math import log
print(briggs(2), log(2))
print(briggs(3), log(3))
```

Exercice 5

```
def brouncker(n):
    s = 0
    for i in range(n):
        s = s + 1 / (2*i+1) / (2*i+2)
    return s

from math import log
print(brouncker(1000), log(2))
```