

CoursRepr?sentationNombres2019-2020

November 24, 2019

0.0.1 Représentation des nombres

0.1 Encodage des entiers naturels

0.1.1 Exemple 1

```
In [3]: def chiffres2nombre(t):  
        """Retourne l'écriture en base 10 d'un entier à partir de  
        la liste de ses chiffres"""  
        n = 0  
        expomax = len(t) - 1  
        for k in range(expomax + 1):  
            n = n + 10 ** (expomax - k) * t[k]  
        return n
```

```
In [4]: chiffres2nombre([7,3,4])
```

```
Out[4]: 734
```

0.2 Exercice 3

```
In [6]: def bits2nombre(t):  
        """Retourne l'entier en base dix représenté  
        par une liste de bits t avec bits de poids fort  
        à gauche"""  
        n = 0  
        for bit in t:  
            n = n * 2 + bit  
        return n
```

0.3 Exercice 4

```
In [8]: def additionBinaire8bits(t1, t2):  
        t3 = [0] * 8  
        retenue = 0  
        for k in range(7, -1, -1):  
            s = t1[k] + t2[k] + retenue  
            t3[k] = s % 2  
            retenue = s // 2  
        return t3
```

```

In [10]: additionBinaire8bits([1,0,1,0,1,1,1,0],[0,0,0,0,1,1,1,1])
Out[10]: [1, 0, 1, 1, 1, 1, 0, 1]
In [2]: bin(37), bin(18)
Out[2]: ('0b100101', '0b10010')
In [3]: import numpy as np
In [5]: np.int8(127) + np.int8(1)
/home/fjunier/.local/lib/python3.6/site-packages/ipykernel_launcher.py:1: RuntimeWarning: over
    """Entry point for launching an IPython kernel.

Out[5]: -128

```

0.4 Représentation des réels

```

In [3]: def decomposition_fraction_egyptienneV1(x):
        """Retourne une liste de dénominateurs entiers p1,, p2, ..., pn
        tels que x = 1/p1 + 1/ p2 + .... + 1 / pn"""
        decomp = []
        while x != 0:
            n = 1
            while 1/n > x:
                n += 1
            decomp.append(n)
            x = x - 1 / n
            print(x, decomp) #pour le débogage
        return decomp

In [4]: import math

        def decomposition_fraction_egyptienneV2(x):
            """Retourne une liste de dénominateurs entiers p1,, p2, ..., pn
            tels que x = 1/p1 + 1/ p2 + .... + 1 / pn"""
            decomp = []
            while x != 0:
                n = math.ceil(1/x)
                decomp.append(n)
                x = x - 1 / n
                print(x, decomp) #pour le débogage
            return decomp

In [9]: decomposition_fraction_egyptienneV1(3/4)
0.25 [2]
0.0 [2, 4]

Out[9]: [2, 4]

```