

Feuille d'exercices : codage des nombres

Représentation des entiers relatifs

Exercice 1 (Un peu culturel) Voici une adresse IPv6, qui représente 8 nombres codés en hexadécimal sur 16 bits : 2001:0db8:3c4d:0015:0000:0000:1a2f:1a2b

1. Donner la valeur de l'entier codé par 0db8.
2. Donner les 4 chiffres hexadécimaux de l'entier 50000, en déduire son écriture binaire.
3. Lire et commenter

```
>>> help(chr)
```

```
Help on built-in function chr in module builtins:
```

```
chr(i, /)
```

```
Return a Unicode string of one character with ordinal i; 0 <= i <= 0x10ffff.
```

Exercice 2 (En vrac) On suppose que les entiers relatifs sont codés en complément à deux sur $n = 8$ bits.

1. Quels entiers relatifs peut-on représenter ?
2. Donner les mots binaires de 8 bits qui représentent les entiers

$0, 1, 2, -1, -2, 127, 126, -127, -128.$

3. Quels sont les entiers relatifs dont la représentation binaire sur 8 bits est :

0000 0000, 1000 0000, 1111 1111, 0111 1111, 0101 0001, 1000 0001

4. Prévoir le résultat de $128 + 3$, puis de $2 * 128$. Vérifier à l'aide du module numpy de Python. On rappelle que `np.int8(x)` code sur 8 bits l'entier relatif x

Représentation des nombres flottants et norme IEEE-754

On suppose dans cette section que les nombres flottants sont codés en 64 bits (on dit parfois double précision) selon la norme IEEE-754, ce qui est le cas de Python 3

Exercice 3 (En vrac)

1. Donner la représentation flottante des nombres suivants :

$$9, -9, 1, \frac{5}{8}, 2^{-53}, 1 + 2^{-52}, 1 + 2^{52}$$

2. Prévoir les tests d'égalité entre les nombres : $1, 1 + 10^{-15}, 1 + 10^{-17}.$
3. Comparer le plus grand entier relatif codé sur 64 bits en complément à deux au plus grand entier représentable par un nombre flottant en 64 bits.
4. Donner un exemple d'entier relatif codé sur 64 bits en complément à deux qui n'est pas représentable par un nombre flottant en 64 bits.